

Spring 5-1-2001

The PREMIUM Club Credit Card Membership Program: Implementation at CAPITAL Card Services

Paul Patterson
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Patterson, Paul, "The PREMIUM Club Credit Card Membership Program: Implementation at CAPITAL Card Services" (2001).
Masters Theses. 7.
<https://scholar.dsu.edu/theses/7>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

The PREMIUM Club Credit Card Membership Program

Implementation at CAPITAL Card Services

Submitted by

Paul Patterson

Prepared for

INFS 790
Information Systems Project

May 10, 2001

CONTENTS

| | |
|--|-----------|
| INTRODUCTION..... | 4 |
| Company Background | 4 |
| Project Description | 5 |
| ANALYSIS | 6 |
| Requirements | 6 |
| Project Plan..... | 11 |
| SYSTEM DESIGN..... | 12 |
| Process Modeling | 13 |
| Logical Database Design..... | 14 |
| Physical Database Design..... | 15 |
| Database Design Language | 17 |
| Appendix A - Central States Indemnity | 21 |
| Appendix B - First Data Resources | 22 |
| Appendix C - Reports | 25 |
| Incentive Report | 25 |
| Enrollment Report | 26 |
| Cancellation Report | 27 |
| Appendix D - Log Files | 28 |
| Enrollment Processing..... | 28 |
| Enrollment from Applications | 29 |
| Premium Club Labels File..... | 30 |
| Appendix E – Source Code..... | 31 |
| Stored Procedures | 31 |
| Premium Club Source Code | 33 |
| Roborep System Source Code: | 42 |
| Caps System source code: | 58 |
| FDR Error codes..... | 75 |
| INDEX | 77 |

LIST OF ILLUSTRATIONS

| | |
|---|----|
| Figure 1 CAPS Main Menu | 6 |
| Figure 2 Premium Club Form | 7 |
| Figure 3 FPX Screen | 7 |
| Figure 4 FPX Enrollment Example..... | 8 |
| Figure 5 Adding Premium Club..... | 8 |
| Figure 6 Account Information Screen..... | 9 |
| Figure 7 Deleting Premium Club | 9 |
| Figure 8 Gantt Chart | 11 |
| Figure 9 Network Diagram | 12 |
| Figure 10 Data Flow Diagram | 13 |
| Figure 11 Entity Relationship Diagram | 14 |

LIST OF TABLES

| | |
|--|----|
| Table 1 Premium Club Form Design | 6 |
| Table 2 Premium Club Reports..... | 10 |
| Table 3 Task Information..... | 11 |
| Table 4 Premium Club Database Metadata | 15 |

INTRODUCTION

Company Background

Sioux Falls, SD based CAPITAL Card Services Inc. (CCSI) is a credit card portfolio management company that provides financial institutions with business strategies for penetrating the credit card market. CCSI offers account management services and operational support that enables banks to expand their credit card portfolios. CCSI offers local and regional banks support and expertise in all areas of credit card administration from credit underwriting to overall portfolio management. CCSI provides the human capital needed to support a growing card business and deliver all components necessary to execute card issuing. CCSI provides the following services:

- ❖ Pre-Processing
 - Mail-room Operations
- ❖ New Accounts
 - Data Entry
 - Credit Decision
 - Credit Correspondence
- ❖ Customer Service
 - Application Solicitation
 - Application Status
 - Correspondence
 - Disputes
 - Chargebacks
 - Security
 - Retention
- ❖ Collections
- ❖ Marketing
 - Pre-Screen Process
 - Direct Mail Marketing
 - Internet Marketing

CAPITAL Card Services Mission:

To provide innovative, comprehensive and effective account management services and products for clients seeking to improve the performance of their bankcard business.

CCSI is privately held by former Governor of South Dakota, Frank Farrar. Mr. Farrar owns a network of financial institutions spanning 5 states including SD, MN, IL, NM and NV. National holdings include banking, mortgage lending and card issuing/servicing operations. The CCSI office is located in the north industrial park area at 500 East 60th Street North in Sioux Falls, South Dakota. The Sioux Falls community and surrounding areas have several banks in the credit card business, which provides an excellent business opportunity for a service organization like CCSI. The local workforce has extensive experience in the financial industry providing the resources necessary to meet operational needs.

Project Description

A project has been approved at CCSI to implement a product called Premium Club. Premium Club is a membership program in which members are eligible for a wide array of products and services designed to simplify the lives of cardholders and save them money. The benefits include:

- ❖ Membership cards for hotels, car rental agencies, eye care providers, and gift-giving agencies.
 - These cards provide discounts of 10-30% and can be used over and over again for continued savings. In addition, members receive special discount coupons for additional savings.
- ❖ Premium Protection for credit cards
- ❖ 48-hour Emergency Card Replacement for lost or stolen cards and access to emergency cash.
- ❖ Exclusive Customer Service
- ❖ Savings on flower and gift basket orders
- ❖ Financial management guide. This guide provides budgeting hints and information about how to use credit cards effectively.
- ❖ Automatic credit line increase reviews
 - Every six months the customer's account is automatically reviewed for a credit line increase.
 - The account is monitored monthly to give an increase as soon as the account qualifies.

The accounts requesting enrollment for the Premium Club program can come from five sources; representatives taking calls in the call-center; applications mailed from potential card members; the IVR (Interactive Voice Response), CSI (Central States Indemnity); future plans are in place for enrollment via the internet using IWR (Interactive Web Response).

To provide the functionality of enrolling and removing accounts from the Premium Club program will involve the modification of two production systems. The first system to be modified is the CAPITAL Application Processing System (CAPS). CAPS provides company-wide information on the credit card applications as they are processed through the system. An area will need to be added to CAPS that can be accessed by customer service representatives to enroll cardholders real-time.

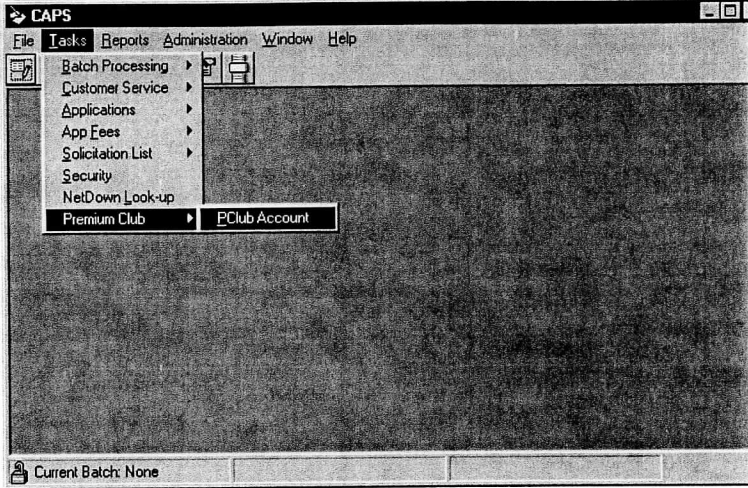
The second system to be modified is named Roborep. Roborep works as a gateway between the in-house database and account information held at First Data Resources (FDR). Roborep makes operations more efficient by simulating keystrokes that a representative would make thereby automating manual processes. Roborep uses the High Level Language Application Programming Interface (HLLAPI) protocol which defines an application programming interface to communicate with and control a terminal emulator. Roborep, a Visual Basic program developed in-house, connects to a mainframe, logs in and initiates various functions on the mainframe. Roborep is able to send keystrokes to the mainframe and receive screen images. Roborep can display the mainframe dialog on the screen to the user. This interface has sometimes been referred to as "screen scraping".

ANALYSIS

Requirements

Objective: Consolidate all Premium Club processes into CAPS to provide the functionality of enrolling and deleting accounts from Premium Club. This will involve moving the current system to CAPS and provide additional information in order to make the process more efficient in the call center (SRF#286).

Table 1 Premium Club Form Design

| | |
|--|--|
| Add a menu option in CAPS for Premium Club (Figure 1) and display the following information on the Premium Club form (Figure 2): | |
| Account number Cardholder name Cardholder address | <p>Figure 1 CAPS Main Menu</p>  |
| Enroll in Premium Club | The representative can enroll the cardholder by clicking in this box. |
| Date Enrolled | Date the cardholder was enrolled in Premium Club. |
| Date Canceled | Date the cardholder cancelled premium club. |
| Account Open Date | Date the account was opened. |
| Last CLI Date | Date of the last credit line increase. |
| Source Information | <p>The representative will select a source of enrollment. Only one box should be checked. The source options are:</p> <ol style="list-style-type: none"> 1) Application - from the original application. 2) Inbound Call - a telephone call in the Call Center. 3) Card Carrier - worked in Correspondence. 4) Solicitation - outbound telemarketing. 5) IVR - those people who choose the enroll option through the IVR. |

Add a menu option in CAPS for Premium Club (Figure 1) and display the following information on the Premium Club form (Figure 2):

| | |
|-----------------------------|---|
| Last Billing Date | The last date the cardholder was charged the fee. |
| Cancel Premium Club | The representative can check this to cancel pclub. |
| Next Billing Date | The next time the cardholder will be billed the Premium Club fee. |
| Reason for Canceling | The representative will place a check mark in the appropriate box next to the reason for canceling. The representative should be allowed to check more than one of these options. The cancel options are: 1) No CLI given. 2) No packet received. 3) Poor service. 4) Coupons not valuable. |

Steps to access account information:

- 1) The representative will key the 16 digit account number on the Premium Club form.
- 2) The system will access FDR for the account information.
- 3) Verify connection to FDR.
 - a) Logon to FDR if no active session is found.
- 4) Verify account number by keying CIS + account number on the FDR screen.
 - a) If the message "Invalid Account Number" appears in position 24/009-24/030, then return a message to the user stating "Invalid Account Number".
- 5) Access the FPX screen by typing FPX + account number (Figure 3).
 - a) Search for 'CLUB'.
 - b) If CLUB is not found, the cardholder is not enrolled.
 - i) No date should be placed in the 'Date Enroll' or 'Last Billing Date' fields.
 - c) If CLUB is found, and positions 006-013 of that line are not populated, then the

Figure 2 Premium Club Form

Figure 3 FPX Screen

enrolled.

- i) A check mark should appear in the 'Enroll in Premium Club' field.
- ii) Proceed down one line and to positions 061-068. Record these 8 digits in the 'Last Billing Date' field.
- iii) Proceed to positions 060-067 of that line. Record these 8 digits in the 'Next Billing Date' field. If the 8 digits equal '99/99/99' then return the word 'Canceled' in the 'Next Billing Date' field.
- d) If CLUB is found, and positions 006-013 of that line are populated, then the cardholder has canceled Premium Club.
 - i) Proceed to position 071-078 of that line, and record the 8 digits in the 'Date Enroll' field.

Figure 4 FPX Enrollment Example

| DELETE ACTN DATE | FEE PROGRAM | GROUP ID | ACCOUNT FEE AMT | ASMT TIMES COUNT | FEE DATE NEXT LAST | RECORD DATE RGSTR UPDATE |
|------------------|-------------|----------|-----------------|------------------|--------------------|--------------------------|
| | \$6MAINT | 00001000 | 9999999999.99 | 999 | 04/09/01 | 03/09/01 |
| A | CLUBSSB | | | 000 | 03/09/01 | 03/09/01 |

Now that the representative has the account information they may enroll or cancel Premium Club or inform the customer on the status of their account.

Steps to enroll an account in Premium Club.

- 1) Check the 'Enroll in Premium Club' box, select a source and click Process.
- 2) CAPS will now access the FPX screen (Figure 4).
 - a) Search for the first available line.
 - b) Key 'A' at column 2 in the first available row.
 - c) Key the Premium Club method at column 16 in the current row.
 - d) Select action key PF2 to update the account.
 - e) Verify the update was successful by checking the message at location row 72 column 2 (Figure 5).
- 3) Switch to the NM CS screen.
 - a) Key 'P' at location row 10 column 50.
 - b) Select Enter key to update the account.

Figure 5 Adding Premium Club

| DELETE ACTN DATE | FEE PROGRAM | GROUP ID | ACCOUNT FEE AMT | ASMT TIMES COUNT | FEE DATE NEXT LAST | RECORD DATE RGSTR UPDATE |
|------------------|-------------|----------|-----------------|------------------|--------------------|--------------------------|
| - | \$6MAINT | 00001000 | 9999999999.99 | 999 | 04/09/01 | 03/09/01 |
| | CLUBSSB | 00002000 | 9999999999.99 | 999 | 03/30/01 | 03/30/01 |
| | | | | 000 | 00/00/00 | 03/30/01 |

- 4) Switch to the NM NA account information screen (Figure 6).
 - a) Capture the name and address information to mail the Premium Club information.
- 5) If the cardholder is re-enrolling, the representative will check the 'Enroll in Premium Club' box, and the current date should replace the date already populated in the 'Date Enrolled' field.
- 6) Record the enrollment in the CAPS database.

Steps to cancel enrollment for an account currently enrolled in Premium Club:

- 7) Access the FPX screen.
 - a) Search for the line containing the original enrollment.
 - b) Key 'D' at column 2 in the enrollment line (Figure 7).
 - c) Select action key PF2 to update the account.
 - d) Verify the update was successful by checking the message at location row 72 column 2.
 - e) Select action key PF11 to verify the update.
- 8) If the representative checks the cancellation box, check if the date of cancellation is less than 6 months from the enroll date, then display the following message "Cardholder needs to return their Premium Club packet in order to cancel." The representative should still be able to cancel pclub, this is only a reminder to them.
- 9) Record the cancellation in the CAPS database.

Steps to process enrollment file from CSI:

- 1) Add menu option to Roborep to accommodate CSI enrollment processing.
- 2) Log on to FDR.
- 3) Read the CSI enrollment file.
- 4) Loop through the file and follow the steps from individual enrollment show above.

Figure 6 Account Information Screen

Figure 7 Deleting Premium Club

| DELETE ACTN DATE | FEE PROGRAM | GROUP ID | ACCOUNT FEE AMT | ASMT TIMES | FEE DATE NEXT LAST | RECORD DATE RGSTR UPDATE |
|------------------|-------------|----------|-----------------|------------|--------------------|--------------------------|
| | FN-FPERR | 99999999 | 9999999999.99 | 999 | 05/16/01 | 02/16/01 |
| D | PCLUB395 | 00002000 | 9999999999.99 | 999 | 04/30/01 | 04/26/01 |

Table 2 Premium Club Reports

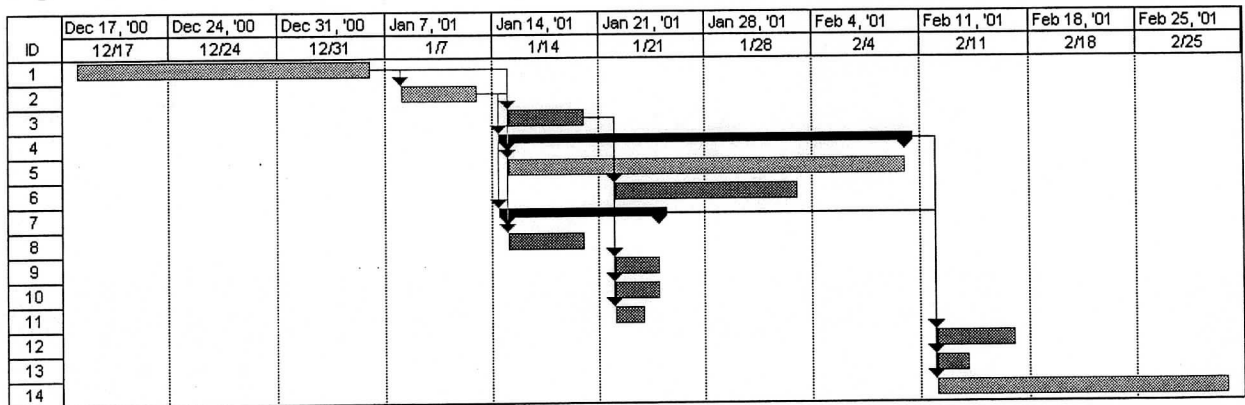
| Report Name | Description |
|--------------------|--|
| Error | Any account enrollments that are entered, but were not able to be enrolled. Include the source of enrollment. |
| Enrollment | An enrollment report to determine incentives. The option of day, month and year and the option of choosing the enrollment source and representative who enrolled the cardholder. This should be sortable by enrollment source, month, or representative. |
| Cancellation | A report listing accounts that have canceled with the option of day, month and year and the cancellation reason. |
| Applications | A report showing the number of applications received each month with PClub marked. Identify those applications that are approved. |

Project Plan

Table 3 Task Information

| ID | Task_Name | Duration | Pred. | Start_Date | Finish_Date | Slack |
|----|------------------------------------|----------|-------|------------|-------------|---------|
| 1 | Requirements Collection | 15 days | | 01-Dec-00 | 21-Dec-00 | 0 days |
| 2 | Process Design | 5 days | 1 | 22-Dec-00 | 28-Dec-00 | 0 days |
| 3 | Database Design | 5 days | 2 | 29-Dec-00 | 04-Jan-01 | 5 days |
| 4 | CAPS Modifications | 20 days | 2 | 29-Dec-00 | 25-Jan-01 | 0 days |
| 5 | Incorporate Host Connectivity | 20 days | 1 | 29-Dec-00 | 25-Jan-01 | 0 days |
| 6 | Design Enrollment Form | 10 days | 3 | 05-Jan-01 | 18-Jan-01 | 5 days |
| 7 | Roborep Modifications | 8 days | 2 | 29-Dec-00 | 09-Jan-01 | 12 days |
| 8 | Create Batch Processing Procedures | 5 days | 2 | 29-Dec-00 | 04-Jan-01 | 15 days |
| 9 | Automate IVR File | 3 days | 3 | 05-Jan-01 | 09-Jan-01 | 12 days |
| 10 | Automate CSI File | 3 days | 3 | 05-Jan-01 | 09-Jan-01 | 12 days |
| 11 | Automate Label Printing | 2 days | 3 | 05-Jan-01 | 08-Jan-01 | 13 days |
| 12 | Report Design | 5 days | 4,7 | 26-Jan-01 | 01-Feb-01 | 10 days |
| 13 | Convert Data from Legacy System | 2 days | 4,7 | 26-Jan-01 | 29-Jan-01 | 13 days |
| 14 | Testing | 15 days | 4,7 | 26-Jan-01 | 15-Feb-01 | 0 days |

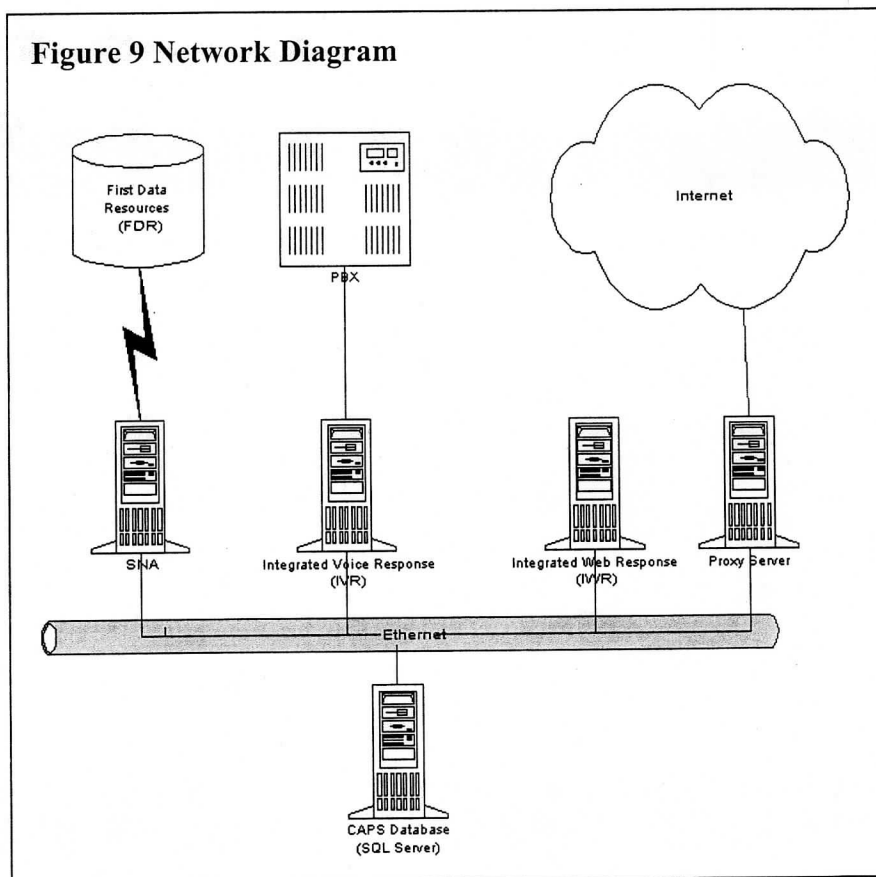
Figure 8 Gantt Chart



SYSTEM DESIGN

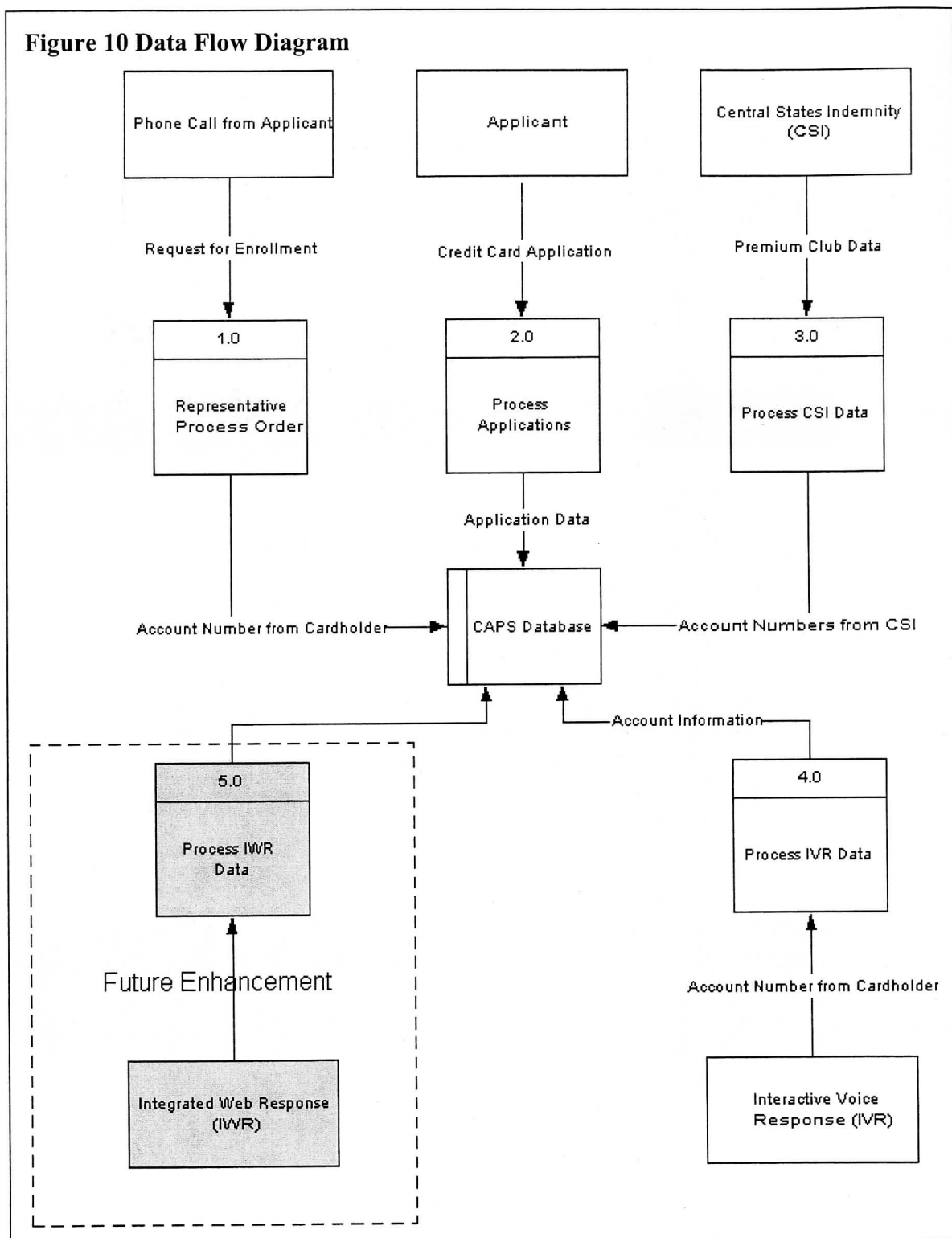
Network Architecture Notes:

- CCSI maintains a connection with FDR using Microsoft SNA server. Client connectivity is established through a product called EXTRA! made by Attachmate.
- CAPS was developed in-house and written entirely in Visual Basic and SQL Server.
- The IVR and IWR run on a system called Vista made by Phoenix, AZ based company Syntellect. Vista is a complete solution for interactive voice response, voice-enabled websites, computer telephony integration, fax on demand and advanced speech recognition.



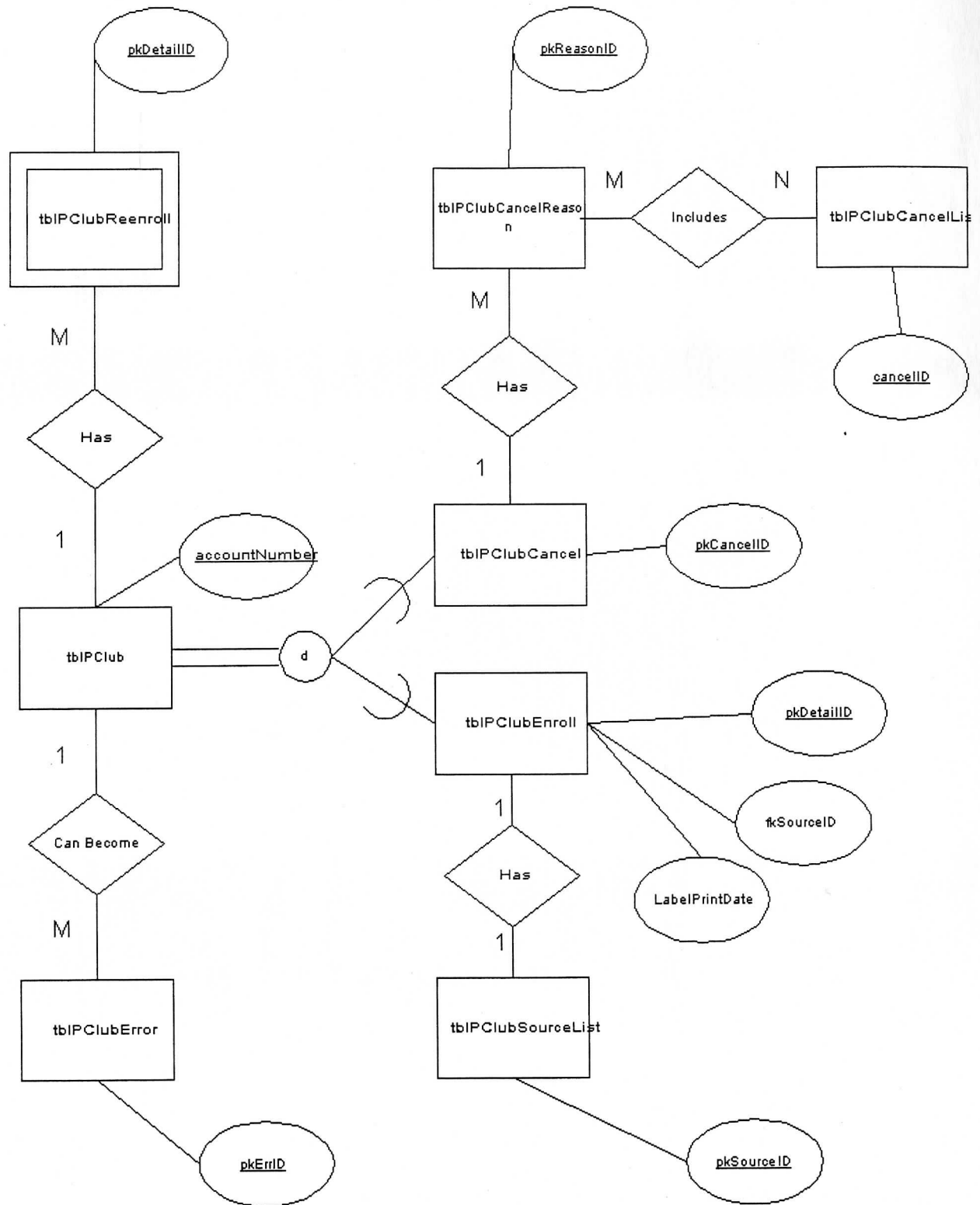
Process Modeling

Figure 10 Data Flow Diagram



Logical Database Design

Figure 11 Entity Relationship Diagram



Physical Database Design

Table 4 Premium Club Database Metadata

| TABLE_NAME | COLUMN_NAME | DATATYPE | PRECISION | LENGTH | NULLABLE |
|-----------------------------|----------------|-----------------------|-----------|--------|----------|
| tblPClub | accountNumber | char | 16 | 16 | NO |
| | source | numeric | 18 | 20 | YES |
| | name | varchar | 25 | 25 | YES |
| | address | varchar | 50 | 50 | YES |
| | city | varchar | 25 | 25 | YES |
| | state | char | 2 | 2 | YES |
| | zip | varchar | 10 | 10 | YES |
| | enrollFlag | bit | 1 | 1 | NO |
| tblPClubCancel | pkCancelID | numeric() identity | 18 | 20 | NO |
| | accountNumber | char | 16 | 16 | YES |
| | userID | varchar | 25 | 25 | YES |
| | cancelDate | smalldatetime | 16 | 16 | YES |
| | | | | | |
| tblPClubCancelList | cancelID | numeric() identity | 18 | 20 | NO |
| | Reason | varchar | 50 | 50 | YES |
| | | | | | |
| tblPClubCancelReason | pkReasonID | numeric() identity | 18 | 20 | NO |
| | fkCancelID | numeric | 18 | 20 | YES |
| | reasonCode | numeric | 18 | 20 | YES |
| | cancelDate | smalldatetime | 16 | 16 | YES |
| | | | | | |
| tblPClubEnroll | pkDetailID | numeric() identity | 18 | 20 | NO |
| | AccountNumber | char | 16 | 16 | YES |
| | FkSourceID | numeric | 18 | 20 | NO |
| | UserID | varchar | 25 | 25 | YES |
| | EnrollDate | smalldatetime | 16 | 16 | NO |
| | LabelPrintDate | smalldatetime | 16 | 16 | YES |
| | | | | | |
| tblPClubError | PkErrID | numeric() identity | 18 | 20 | NO |
| | AccountNumber | char | 16 | 16 | YES |
| | ErrTime | smalldatetime | 16 | 16 | YES |
| | UserID | char | 10 | 10 | YES |
| | ErrMsgage | varchar | 100 | 100 | YES |

| TABLE_NAME | COLUMN_NAME | DATATYPE | PRECISION | LENGTH | NULLABLE |
|---------------------------|---------------|-----------------------|-----------|--------|----------|
| | | | | | |
| tblPCLubReenroll | PkRecordID | numeric() identity | 18 | 20 | NO |
| | AccountNumber | char | 16 | 16 | YES |
| | RequestDate | smalldatetime | 16 | 16 | YES |
| | FkSourceID | numeric | 18 | 20 | YES |
| | UserID | varchar | 25 | 25 | YES |
| | ReenrollDate | smalldatetime | 16 | 16 | YES |
| | | | | | |
| tblPCLUBMissed | AccountNumber | nvarchar | 17 | 34 | NO |
| | Status | nvarchar | 25 | 50 | YES |
| | | | | | |
| tblPCLubSourceList | PkSourceID | numeric() identity | 18 | 20 | NO |
| | Source | varchar | 25 | 25 | YES |
| | | | | | |
| | | | | | |

Database Design Language

```
CREATE TABLE [dbo].[tblPCLub] (  
    [accountNumber] [char] (16) NOT NULL ,  
    [source] [numeric](18, 0) NULL ,  
    [name] [varchar] (25) NULL ,  
    [address] [varchar] (50) NULL ,  
    [city] [varchar] (25) NULL ,  
    [state] [char] (2) NULL ,  
    [zip] [varchar] (10) NULL ,  
    [enrollFlag] [bit] NOT NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[tblPCLubCancelList] (  
    [cancelID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,  
    [Reason] [varchar] (50) NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[tblPCLUBMissed] (  
    [AccountNumber] [nvarchar] (17) NULL ,  
    [Status] [nvarchar] (25) NULL  
)
```

```
CREATE TABLE [dbo].[tblPCLubSourceList] (  
    [pkSourceID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,  
    [Source] [varchar] (25) NULL  
) ON [PRIMARY]
```

```
CREATE TABLE [dbo].[tblProductMatrix] (  
    [sourceCode] [varchar] (4) NOT NULL ,  
    [productCode] [varchar] (4) NULL ,  
    [agent] [char] (4) NULL ,  
    [cardPricingStrategy] [char] (4) NULL ,  
    [monthlyParticipationFee] [char] (8) NULL ,  
    [processingFee] [money] NULL ,  
    [expressProcessingFee] [money] NULL ,  
    [addCardMethod] [char] (8) NULL ,  
    [counterOfferSourceCode] [varchar] (4) NULL ,  
    [system] [varchar] (4) NULL ,  
    [principle] [varchar] (4) NULL ,  
    [prefix] [varchar] (6) NULL ,  
    [ProductDefinition] [varchar] (4) NULL ,  
    [clientNumber] [varchar] (4) NOT NULL ,  
    [location] [char] (6) NOT NULL ,  
    [clientBank] [char] (1) NULL ,
```

```

    [PClubMethod] [varchar] (8) NULL ,
    [Marketer] [char] (10) NULL ,
    [CntrOffrLetter] [varchar] (10) NULL ,
    [securityDepositMinBal] [money] NULL ,
    [productType] [varchar] (25) NULL
) ON [PRIMARY]

```

```

CREATE TABLE [dbo].[tblPClubCancel] (
    [pkCancelID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [accountNumber] [char] (16) NULL ,
    [userID] [varchar] (25) NULL ,
    [cancelDate] [smalldatetime] NULL
) ON [PRIMARY]

```

```

CREATE TABLE [dbo].[tblPClubDetail] (
    [pkDetailID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [accountNumber] [char] (16) NULL ,
    [fkSourceID] [numeric](18, 0) NULL ,
    [userID] [varchar] (25) NULL ,
    [enrollDate] [smalldatetime] NULL ,
    [labelPrintDate] [smalldatetime] NULL
) ON [PRIMARY]

```

```

CREATE TABLE [dbo].[tblPClubError] (
    [pkErrID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [accountNumber] [char] (16) NULL ,
    [errTime] [smalldatetime] NULL ,
    [userID] [char] (10) NULL ,
    [errMessage] [varchar] (100) NULL
) ON [PRIMARY]

```

```

CREATE TABLE [dbo].[tblPClubReenroll] (
    [pkRecordID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [accountNumber] [char] (16) NULL ,
    [requestDate] [smalldatetime] NULL ,
    [fkSourceID] [numeric](18, 0) NULL ,
    [userID] [varchar] (25) NULL ,
    [reenrollDate] [smalldatetime] NULL
) ON [PRIMARY]

```

```

CREATE TABLE [dbo].[tblPClubCancelReason] (
    [pkReasonID] [numeric](18, 0) IDENTITY (1, 1) NOT NULL ,
    [fkCancelID] [numeric](18, 0) NULL ,
    [reasonCode] [numeric](18, 0) NULL ,
    [cancelDate] [smalldatetime] NULL
) ON [PRIMARY]

```

CREATE CLUSTERED INDEX [index_appFeeID] ON [dbo].[tblAppFees]([appFeeID])
WITH FILLFACTOR = 90 ON [PRIMARY]

CREATE CLUSTERED INDEX [seqNumber_Index] ON [dbo].[tblNetDown]([seqNumber])
WITH FILLFACTOR = 90 ON [PRIMARY]

CREATE CLUSTERED INDEX [AcctNo] ON [dbo].[tblPCLUBMissed]([AccountNumber])
WITH FILLFACTOR = 90 ON [PRIMARY]

CREATE CLUSTERED INDEX [appIDIndex] ON [dbo].[tblApplications]([appID]) WITH
FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLub] WITH NOCHECK ADD
CONSTRAINT [DF_tblPCLub_enrollFlag] DEFAULT (0) FOR [enrollFlag],
CONSTRAINT [PK_tblPCLub] PRIMARY KEY NONCLUSTERED
([accountNumber]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubCancelList] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubCancelList] PRIMARY KEY NONCLUSTERED
([cancelID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubSourceList] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubSourceList] PRIMARY KEY NONCLUSTERED
([pkSourceID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubCancel] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubCancel] PRIMARY KEY NONCLUSTERED
([pkCancelID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubDetail] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubDetail] PRIMARY KEY NONCLUSTERED
([pkDetailID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubError] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubError] PRIMARY KEY NONCLUSTERED
([pkErrID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubReenroll] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubReenroll] PRIMARY KEY NONCLUSTERED
([pkRecordID]) WITH FILLFACTOR = 90 ON [PRIMARY]

ALTER TABLE [dbo].[tblPCLubCancelReason] WITH NOCHECK ADD
CONSTRAINT [PK_tblPCLubCancelReason] PRIMARY KEY NONCLUSTERED
([pkReasonID]) WITH FILLFACTOR = 90 ON [PRIMARY]


```
CREATE INDEX [Index_AccountNumber] ON [dbo].[tblPClubError] ([accountNumber]) ON  
[PRIMARY]
```

```
CREATE INDEX [Index_AccountNumber] ON [dbo].[tblPClubDetail] ([accountNumber]) ON  
[PRIMARY]
```

```
ALTER TABLE [dbo].[tblPClubCancel] ADD  
    CONSTRAINT [FK_tblPClubCancel_tblPClub] FOREIGN KEY  
    ([accountNumber]) REFERENCES [dbo].[tblPClub] ([accountNumber])
```

```
ALTER TABLE [dbo].[tblPClubDetail] ADD  
    CONSTRAINT [FK_tblPClubDetail_tblPClub] FOREIGN KEY  
    ([accountNumber]) REFERENCES [dbo].[tblPClub] ([accountNumber] ),  
    CONSTRAINT [FK_tblPClubDetail_tblPClubSourceList] FOREIGN KEY  
    ([fkSourceID]) REFERENCES [dbo].[tblPClubSourceList] ([pkSourceID])
```

```
ALTER TABLE [dbo].[tblPClubError] ADD  
    CONSTRAINT [FK_tblPClubError_tblPClub] FOREIGN KEY  
    ([accountNumber]) REFERENCES [dbo].[tblPClub] ([accountNumber] )
```

```
ALTER TABLE [dbo].[tblPClubReenroll] ADD  
    CONSTRAINT [FK_tblPClubReenroll_tblPClub] FOREIGN KEY  
    ([accountNumber]) REFERENCES [dbo].[tblPClub] ([accountNumber] )
```

```
ALTER TABLE [dbo].[tblPClubCancelReason] ADD  
    CONSTRAINT [FK_tblPClubCancelReason_tblPClubCancel] FOREIGN KEY  
    ([fkCancelID]) REFERENCES [dbo].[tblPClubCancel] ([pkCancelID]),  
    CONSTRAINT [FK_tblPClubCancelReason_tblPClubCancelList] FOREIGN KEY  
    ([reasonCode]) REFERENCES [dbo].[tblPClubCancelList] ([cancelID] )
```

Appendix A - Central States Indemnity

Central States Indemnity (CSI) is a specialty insurance company with a niche in the bankcard market. CSI is a provider of Payment Protection Programs to financial institutions across the country. CSI's products are designed to help customers with their monthly payments in the event of death, disability, involuntary unemployment, or an unpaid leave of absence to care for a family member. Services include a total solution for marketing and administering payment protection programs. CSI specializes in payment protection programs designed to help consumers with their monthly bills. These plans of insurance are made available on a group basis to customers of participating credit card issuers. Payment Protection Plans are customized to suit preferences or requirements.

CSI programs protect covered cardholders' account balances by making payments to their accounts when they cannot. CSI's programs to now include:

- ❖ Administrative Services and/or Marketing support for Debt Cancellation/Deferment products.
- ❖ Insurance plans for the Retail Marketplace which include life, disability, unemployment, family leave, property and job retraining benefits.
- ❖ Non-Credit Payment Protection Products, where benefits are paid "directly" to the cardholder.

Appendix B - First Data Resources

First Data Resources Inc. (FDR) located in Omaha Nebraska applies the resources of advanced technology to help meet the business goals CAPITAL's transaction card plan. FDR can enhance the profitability of CCSI's transaction card operation by bringing a customized processing package. FDR System features can meet your transaction card plan's marketing goals through the following:

- ❖ Merchandise and service advertisements you select for mailing with statements, computer-generated correspondence, and plastics.
- ❖ Merchant and cardholder account records automatically selected from FDR's computer files according to criteria you select (mailing labels are also available).
- ❖ Processing enhancements that keep your operation on the leading edge in the transaction card marketplace.

FDR supports cardholder service efforts with online capabilities for cardholder applications, new account entry, and updating information on existing cardholder account records. In addition, cardholder statements and account balance and status information can be displayed online for easy access by your customer service representatives.

- ❖ **Cardholder Applications** - To process cardholder applications for transaction cards, you can use FDR's Application System.
 - Enter cardholder applications online and have them evaluated automatically.
 - Reject them, keep them in a separate online file for future re-evaluation, or approve them and have cardholder account records created for them automatically.
- ❖ **Application Entry** - Information from cardholder application forms can be entered directly into the System by using online transactions.
 - Records for pending and rejected applications are stored separately from cardholder account records until they are either approved or deleted.
 - Approved applications automatically become cardholder account records.
- ❖ **Credit Scoring System** - An applicant's credit history is one factor in determining whether the application for a transaction card is accepted or rejected.
 - Streamline your cardholder application process because credit-evaluation decisions are standardized by parameters you specify.
 - Change parameters as needed.
 - Automatically notify rejected applicants and welcome accepted applicants with customized, computer-generated letters.
 - Exchange credit information with a credit bureau.
- ❖ **New Cardholder Accounts** - You can have new cardholder account records either created automatically whenever applications are approved or manually entered directly into the system in a separate step from your application process. Both of these options allow you to also have customized welcome letters automatically generated and sent to new cardholders.
- ❖ **Account Transfers** - You can establish a new cardholder account record for an existing cardholder who reports a plastic lost or stolen by using FDR's account transfer feature. With this feature, you can automatically transfer existing monetary and non-monetary information to the new account record. If you use the automatic item transfer option of this feature, you can assure that late-posting items also transfer automatically.

- ❖ **Cardholder Plastics and PINs** - You can automatically control the encoding and embossing of plastics through FDR. Available controls affect such factors as the following.
 - ❖ The type and number of plastics your cardholders receive.
 - ❖ Whether the System encodes personal identification numbers (PINs) on plastics.
 - ❖ The type of mailer used to communicate PINs to cardholders.
 - ❖ When to reissue plastics.
- ❖ **Billing and Statements** - Billing options in the FDR System provide control over the volume of cardholder inquiries by matching the production of statements to your customer service staffing plan. With these options, you can have statements produced during approximately the same time each month so that you receive most cardholder inquiries at approximately the same time. Conversely, if you want to receive cardholder inquiries throughout the month, you can have statements produced at various times. You can customize statements sent to your cardholders to suit your needs. The System produces them at cycle time and under such circumstances as when an account becomes overlimit or an account transfer occurs.
- ❖ **Payments** - You can enter cardholder payments into the System from terminals at your location, from FDR, or from a tape. In addition to normal payment processing, the FDR System can support your service to cardholders by automatically withdrawing credit card payments from their checking or savings accounts. You can also offer your cardholders such payment options as those listed below.
 - 90 or 180 days same as cash.
 - Skip payment during a specified month or months.
 - Pay ahead.
- ❖ **Disputes** - If a cardholder disputes an item, you can place that item in dispute in the System. Options available in the FDR System allow you to process disputed items in one of the two following ways:
 - You can remove the amount in dispute from the outstanding balance along with its billed interest. This interest is not recovered.
 - You can have the amount of the disputed item remain in the outstanding balance. The System charges or credits interest after you resolve the dispute according to guidelines you establish.
- ❖ **Retrievals** - When a cardholder questions an item on a statement, the original ticket or a copy of the item can be requested through the FDR Retrieval System. Retrievals are requested by using either online transactions in the Retrieval System or the Customer Inquiry System.
- ❖ **Online References for Cardholder Inquiries** - Your customer service personnel have online access to cardholder balance and status information through the FDR System. This information includes details such as the following.
 - The account's delinquency or overlimit status throughout the last 12 months.
 - When the account cycles each month and its current balance.
 - The amount of available credit and the credit limit.
 - Your customer service personnel can also have online access to cardholder statements if you use FDR's Customer Inquiry System. With this access, they can respond to cardholders' questions about their bills without having to sort through paper and microfiche files of statements.
 - In addition to online statements, the Customer Inquiry System allows you to do the following.

- Keep notes on individual cardholder accounts online.
- Display a cardholder's payment history.
- Request a retrieval or create an outgoing chargeback.
- ❖ **Correspondence** - You can reduce the amount of time required to prepare form letters by using FDR's Computer Letters System. This feature allows you to place customized letters on file in the System and have them produced and mailed automatically in response to changes in account conditions.
- ❖ **Exceptions** - Cardholder accounts that cannot be processed normally are processed as exceptions in the FDR System. Examples of exceptions include a lost or stolen plastic or account delinquency.
- ❖ **Collections** - Payments may need to be collected from cardholders who exceed their credit limit or who are not current with payments. You determine the conditions an account must meet before it automatically enters the Collection System. Delinquency and having charges in excess of the credit limit are examples of these conditions. The Collection System allows you to assign collectors and set up online queues for various types of collection accounts according to your needs.

Appendix C - Reports

Incentive Report

Description: This report is used for the Premium Incentive program for customer service representatives

PClub Incentive Report
For the Period: 3/1/01 to 3/22/01
Enrollment Source: All

| | |
|---------------|------------|
| aburr | 3 |
| asharif | 3 |
| ccox | 5 |
| cmcmilli | 7 |
| cmeyer | 2 |
| CSI | 87 |
| dhighstr | 4 |
| iblakey | 2 |
| jarroyo | 1 |
| jcoughli | 12 |
| jeilmes | 4 |
| jelse | 2 |
| ygloe | 1 |
| ygoldhor | 1 |
| yjohnson | 1 |
| yquezada | 5 |
| yriley | 1 |
| kmccuen | 1 |
| lcavros | 3 |
| lngoran | 1 |
| mbartz | 1 |
| mjess | 1 |
| mmiss | 3 |
| mwells | 3 |
| natwood | 2 |
| pbrandt | 16 |
| rgordy | 1 |
| tbosch | 5 |
| tburr | 1 |
| tdelfs | 1 |
| tmurray | 2 |
| IVR | 21 |
| zwagner | 6 |
| Total: | 209 |

Enrollment Report

Description: This report is used to determine enrollments from various sources based on date range.

PClub Enrollment Report

For the Period 3/1/01 to 4/9/01

Enrollment Source: All

| | |
|---------------------|------------|
| Card Carrier | 501 |
| Inbound | 121 |
| Insert | 3 |
| Outbound | 1 |
| VRU | 90 |
| Grand Total: | 716 |

Cancellation Report

Description: This report is used to determine enrollments from various sources based on date range.

PClub Cancellation Report

For the Period 3/1/01 to 4/9/01
Cancel Reason: All

| | |
|----------------------|------------|
| Coupons not Valuable | 24 |
| No CLI Given | 11 |
| No Packet Received | 59 |
| Servicing | 96 |
| Grand Total | 190 |

Appendix D - Log Files

Enrollment Processing

Purpose: The log file generated when the Premium Club enrollment from CSI is completed.

```
*****
Results for Premium Club enrollment.
\\PDC\is\VB\RoboRep2\logs\PClub\PClubFromCSI_022200_112306.txt
*****
```

Started upload on: 1/22/01 8:50:55 AM

Errd Account Numbers:

| Account Number | Source | Error Message |
|------------------|---|---------------|
| 4610078004169371 | *** UNAUTHORIZED SYS/PRIN ACCESS *** | |
| 4239801003214448 | Enroll Success! | |
| 4239801003220163 | Enroll Success! | |
| 4239801003222003 | Enroll Success! | |
| 4239801003222904 | Enroll Success! | |
| 4239801003223514 | Enroll Success! | |
| 4239801003225055 | Lost/Stolen Action Code: 4239801003903339 | |
| 4239801003225055 | Enroll Success! | |
| 4239801003245384 | Enroll Success! | |
| 4239801003252455 | Enroll Success! | |
| 5433601170000955 | Enroll Success! | |
| 5433601100085191 | Enroll Success! | |
| 4239801002048987 | Enroll Success! | |
| 4427220140521098 | Enroll Success! | |
| 4545645454546541 | INVALID ACCOUNT | |
| 4071930140271044 | *** UNAUTHORIZED SYS/PRIN ACCESS *** | |
| 4239801003091044 | Enroll Success! | |
| 4239801003091044 | Already Enrolled | |
| 4239801102006265 | Enroll Success! | |
| 4239801003082688 | Enroll Success! | |
| 4239801002378467 | Enroll Success! | |
| 4239801002380919 | Enroll Success! | |
| 4239801002384705 | Enroll Success! | |
| 4239801002386650 | Enroll Success! | |
| 4239801002389597 | Enroll Success! | |
| 4239801002389654 | Enroll Success! | |
| 4239801002389779 | Enroll Success! | |
| 4239801002902811 | Enroll Success! | |

Upload finished on : 2/22/01 9:16:14 AM

Records Processed: 102
Total Errors: 4

Finished Premium Club enrollment process.

Enrollment from Applications

Purpose: The log file generated when the Premium Club enrollment process from applications is completed.

```
*****
Results for PClub processing from applications
\\PDC\is\VB\RoboRep2\logs\PClub\PClubFromApps_022201_112306.txt
*****
```

Started PClub processing at: 02/02/01 11:23:06 AM
Records to process: 77

| | |
|------------------|-------------------|
| 4239801003124324 | Update Successful |
| 4239801003124340 | Update Successful |
| 4239801003124373 | Update Successful |
| 4239801003124431 | Update Successful |
| 4239801003124464 | Update Successful |
| 4239801003124472 | Update Successful |
| 4239801003124548 | Update Successful |
| 4239801003125479 | Update Successful |
| 4239801003125503 | Update Successful |
| 4239801003125552 | Update Successful |
| 4239801003125644 | Update Successful |
| 4239801003125651 | Update Successful |
| 4239801003125669 | Update Successful |
| 4239801003125735 | Update Successful |
| 4239801003125776 | Update Successful |
| 4239801003125800 | Update Successful |
| 4239801003125842 | Update Successful |
| 4239801003126105 | Update Successful |
| 4239801003126113 | Update Successful |
| 4239801003126121 | Update Successful |
| 4239801003126162 | Update Successful |
| 4239801003126220 | Update Successful |
| 4239801003126253 | Update Successful |
| 4239801003126287 | Update Successful |
| 4239801003126303 | Update Successful |
| 4239801003126360 | Update Successful |
| 4239801003126402 | Update Successful |
| 4239801003126436 | Update Successful |
| 4239801003126451 | Update Successful |
| 4239801003126527 | Update Successful |
| 4239801003126568 | Update Successful |
| 4239801003126659 | Update Successful |
| 4239801003126741 | Update Successful |
| 4239801003126774 | Update Successful |
| 4239801003126816 | Update Successful |
| 4239801003126857 | Update Successful |
| 4239801003126881 | Update Successful |
| 4239801003126931 | Update Successful |
| 4239801003126956 | Update Successful |

Finished PClub processing on 02/22/01 11:33:49 AM

Premium Club Labels File

Pclub label file
\\PDC\is\VB\RoboRep2\logs\PClub\Labels\PClubLabels_022201_112306.txt

Guy J Cummings, 20 Haver Hills Apt 306 , Brockton, MA, 02301-6708
Charles Lehtinen, 1232 University Ave , Bakersfield, CA, 93305-1540
Youlondia D Wright, 1652 Garcia Drive Apt B, Ft Gordon, GA, 30905-2813
Michael J Januska, 7 West Pompano Dr , Brick, NJ, 08723-7621
Shawn M Killingback, 223 Dolphin Point Rd # 7, Clearwater, FL, 33767
Mary Childs, 4828 Buckingham Ave , Detroit, MI, 482243545
Rubin Jackson, 5007 Quebec Blvd , Richmond, TX, 774692627
Danieta E Pertile, 412 Mariana Ave , Midland, TX, 797015809
Terrence A Smith, 17394 Valencia Ave # 33, Fontana, CA, 92335-0102
Gerald P Newberry, 12587 Se River Rd # 350, Milwaukie, OR, 97222-8012
Kevin J Bricker, 3887 Evans Dr , Grove City, OH, 431232809
Alvaro H Leighton, 2636 Sw 32nd St , Cape Coral, FL, 33914-4756
Jesse Fox, 8411 Shadow Ct , Coral Springs, FL, 330717469
Robert W Anecchini, 1211 7th Eet Rd , Corbin, KY, 40701
Petra T Montano, Po Box 2171 , West Sacramento, CA, 95691-7171
Beverly L Fuller, Po Box 491 , Apache Junction, AZ, 85217-0491
Rebecca H Brown, Po Box 2736 , Lebanon, VA, 24266-2736
Mark A Mitchell, 1163 Washington Blvd , Baltimore, MD, 212301824
Cheryl M Gibson, 4520 Nw 36th St # 212, Fort Lauderdale, FL, 33319-6425
Richard Charlton, 48 Dogwood Court Apt G, Ashville, NC, 28805
Steve L Dozier, 5014 Denning St , Las Vegas, NV, 89122-7160
Peg F Forthier, 1300 Harding Way # B4, Galion, OH, 44833
Roxana Jimenez, 4528 Nautalis Cir Apt 210 , Fort Worth, TX, 761062462
Melva J Wright, Po Box 25163 , Oklahoma City, OK, 731250163
Robin M Markowitz, 2615 Chestnut Ridge Dr , Pittsburgh, PA, 152054726
Jerald J Woods Jr, 1205 E Bell Ave # 20, Des Moines, IA, 503157438
Fidencio Preciado, 8611 Ivy St 104d, Los Angeles, CA, 900021535
Tammy Dudley, 1801 Ethridge Ct , Olympia, WA, 98506
Josiah Shiel, 122 W 27th St , New York, NY, 100016227
Jacinta Abner, 31 Walden Ave , Buffalo, NY, 142112214
Regenia S Bowen, 1303 Graymont Dr , Atlanta, GA, 30310
Eddie L Loche, 1406 Tulane Dr # 50, Lufkin, TX, 759014742
Troy L Hughes, 2643 Custer Dr , Dallas, TX, 752166553
Emma Young, 8743 Village Green Ct , Alexandria, VA, 223091531
Angela D Rich, Po Box 961205 , Riverdale, GA, 302966902
Charles S Manns, 775 E Broad St , Columbus, OH, 432051007
Lateefah Bilal, 631 Waterloo Rd , Warrenton, VA, 201863015
Luther Shows, 2041 Edwin Ave , Pearl, MS, 392086221
Joseph St Amand, 70 River St , Sanford, ME, 040732839
Michelle Wilson, 556 E 88th St # 2f, Chicago, IL, 606196854
Barbara Love, 868 Shadowridge Dr , Atlanta, GA, 30316
Roberto Rivera, 120 S 8th St # C, Vineland, NJ, 083604751
Bruce Irving, 2123 Rochelle Pl , Columbus, OH, 432323920
Ellis M Graham, 202 Island Ave # 221, San Diego, CA, 921016826

Appendix E – Source Code

Stored Procedures

```
-----
-- PROCEDURE:      sp_CheckExistingPClubAcct
-- PROGRAMMER:     Paul Patterson
-- CREATED ON:     1/19/2001
-- PURPOSE:       Determines if an account number exists in the PClub database.
-- PRE:          tblPClub must exist.
-- POST:         Account Number is returned.
-----
```

```
CREATE PROCEDURE sp_CheckExistingPClubAcct (@AcctNumber VARCHAR(16)) AS
SELECT
    accountNumber
FROM
    tblPClub
WHERE
    accountNumber = @AcctNumber
GO
-----
```

```
-----
-- PROCEDURE:      sp_GetApplicationBasedOnASN
-- PROGRAMMER:     Paul Patterson
-- LAST MODIFIED:   04/19/2000
-- PURPOSE:
-- PRE:
-- POST:
-----
```

```
CREATE PROCEDURE sp_GetApplicationBasedOnASN (@ASN VARCHAR(16)) AS
SELECT
    *
FROM
    tblApplications
WHERE
    asn = @ASN
GO
-----
```

```
-----
-- PROCEDURE:      sp_GetPClubMethod
-- PROGRAMMER:     Paul Patterson
-- LAST MODIFIED:   01/18/2001
-- PURPOSE:       Selects the PClub method from tblProductMatrix
-- PRE:          tblProductMatrix must exist.
-- POST:         PClub method is returned.
-----
```

```
CREATE PROCEDURE sp_GetPClubMethod (@SourceCode VARCHAR(4)) AS
SELECT
    PClubMethod
FROM
    tblProductMatrix
WHERE
    sourceCode = @SourceCode
GO
-----
```

```
-----
-- PROCEDURE:      sp_CheckASN
-- PROGRAMMER:     Paul Patterson
-- CREATED ON:     03/24/2000
-- PURPOSE:       Determines if an ASN already exists.
-- PRE:          tblNetDown must exist.
-- POST:         ASN is returned.
-- NOTES:
-----
```

```
CREATE PROCEDURE sp_CheckASN (@ASN VARCHAR(12)) AS
SELECT
    ASN
FROM
    tblNetDown
WHERE
    ASN = @ASN
GO
-----
```

```

-----
-- PROCEDURE:      sp_GetProductMatrixTable
-- PROGRAMMER:     Paul Patterson
-- CREATED ON:     03/29/2000
-- PURPOSE:       Selects the contents of the product matrix table.
-- PRE:           tblProductMatrix must exist.
-- POST:          Table contents are returned.
-----

```

```

CREATE PROCEDURE sp_GetProductMatrixTable AS
SELECT
    *
FROM
    tblProductMatrix
ORDER BY
    SourceCode
GO

```

```

-----
-- PROCEDURE:      sp_GetSourceCode
-- PROGRAMMER:     Paul Patterson
-- LAST MODIFIED:   04/13/2000
-- PURPOSE:
-- PRE:
-- POST:
-- NOTES:
-----

```

```

CREATE PROCEDURE sp_GetSourceCode (@SequenceNumber VARCHAR(9)) AS
SELECT
    sourceCode
FROM
    tblNetDown
WHERE
    seqNumber = @SequenceNumber
GO

```

```

-----
-- PROCEDURE:      sp_GetSystem
-- PROGRAMMER:     Paul Patterson
-- LAST MODIFIED:   03/24/2000
-- PURPOSE:       Selects the system code from tblProductMatrix.
-- PRE:           tblProductMatrix must exist.
-- POST:          System code is returned.
-- NOTES:         @SourceCode is the original source code prior to the counter offer
--                process.
-----

```

```

CREATE PROCEDURE sp_GetSystem (@SourceCode VARCHAR(4)) AS
SELECT
    [system]
FROM
    tblProductMatrix
WHERE
    sourceCode = @SourceCode
GO

```

```

-----
-- PROCEDURE:      sp_GetAccountNumber
-- PROGRAMMER:     Paul Patterson
-- LAST MODIFIED:   09/07/2000
-- PURPOSE:       Selects the account number from the A20 queue table.
-- PRE:           The ASN of the account
-- POST:          Account Number is returned.
-----

```

```

CREATE PROCEDURE sp_GetAccountNumber (@ASN CHAR(12)) AS

SELECT
    accountNumber
FROM
    tblA20QueueAll
WHERE
    asn = @ASN union
SELECT
    accountNumber
FROM
    TblCOPS WHERE asn = @ASN

```

Premium Club Source Code

Description: This is the source code added to the CAPS system to accommodate the Premium Club program.

```
'-----
' Sub:           UpdatePClubForm
' Programmer:    Paul Patterson
' Last Modified: 01/04/2001
' Purpose:       Update the PClub form with the appropriate image and message.
' Pre:           The image to show and the message text to display.
' Post:          The form will be updated.
'-----

Public Sub UpdatePClubForm(ByVal ImageX As Image, Optional ByVal Message As String)
On Error GoTo updateErr

    Dim imgX As Image
    Set imgX = ImageX

    'Hide all images.
    With Me

        .imgEnrolled.Visible = False
        .imgNotEnrolled.Visible = False
        .imgError.Visible = False
        .imgWait.Visible = False
        .imgFDR.Visible = False

    End With

    'Show the select image.
    If Not ImageX Is Nothing Then imgX.Visible = True

    'Update the message.
    If Not IsMissing(Message) Then frmPClub.lblMessage = Message

Exit Sub

EXIT_PROC:
Exit Sub

updateErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Resume EXIT_PROC

End Sub
```

```

-----
' Sub:                ProcActivity
' Programmer:         Paul Patterson
' Created on:         12/21/2000
' Purpose:            This will validate the form and determine which
'                    action to take on the account.
' Pre:
' Post:
-----

Sub ProcActivity()
On Error GoTo ActivityErr

    Dim dteEnrolled As Date
    Dim dteCanceled As Date

    'No boxes checked
    If Me.chkEnrolled = 0 And Me.chkCancelPClub = 0 Then
        MsgBox "You must check a box to either enroll or cancel.", vbExclamation
        Exit Sub
    End If

    'Both boxes checked.
    If Me.chkEnrolled = 1 And Me.chkCancelPClub = 1 Then
        MsgBox "Please select one option.", vbExclamation
        Exit Sub
    End If

    'Process PClub application based on type.
    Select Case objPClub.PClubType
        Case 0 'Enrolled

            'Customer is enrolled.
            If objPClub.PClubType = btEnrolled Then

                'Cancel box must be checked to process.
                If Me.chkEnrolled = 1 And Me.chkCancelPClub = 0 Then
                    MsgBox "The customer is already enrolled. Please select cancel to continue.",
                        vbExclamation
                    Exit Sub
                End If

                'Verify reason box is selected.
                If VerifyCancelReason = False Then
                    MsgBox "Please select a reason for the cancellation.", vbExclamation
                    Me.lstCancel.SetFocus
                    Exit Sub
                End If

                'Get the date enrolled.
                If Trim(Me.txtDateEnrolled) = "" Then
                    Exit Sub
                Else
                    dteEnrolled = Me.txtDateEnrolled
                End If

                'Get the date canceled.
                If Trim(Me.txtDateCanceled) = "" Then
                    dteCanceled = Date
                Else
                    dteCanceled = Me.txtDateCanceled
                End If

                'Test to see if the cancel date is less than 6 months from enroll date.
                If DateDiff("m", dteCanceled, dteEnrolled) < 6 Then
                    If MsgBox("Cardholder needs to return their Premium Club packet in order to
                        cancel. Do you want to continue cancellation?", vbYesNo) = vbNo Then
                        Call frmPClub.UpdatePClubForm(frmPClub.imgEnrolled, "Cancellation has been
                            aborted...")
                        Exit Sub
                    End If
                End If
            End If
        End Select
    End Sub

```

```

End If      'If DateDiff("m", dteCanceled, dteEnrolled) < 6 Then

'Add or update the PClub record.
If objPClub.CancelPClub(objPClub.AccountNumber) = False Then
    MsgBox "Error encountered during cancellation.", vbExclamation
    Call objPClub.AppendError(objPClub.AccountNumber, "procActivity: " &
        Left(Err.Description, 75))
    Exit Sub
End If

Exit Sub

End If      'If objPClub.PClubType = btEnrolled Then

Case 1 'Cancelled

'Customer is not enrolled or has cancelled enrollment.
If objPClub.PClubType = btCancelled Then

    If Me.chkEnrolled = 0 And Me.chkCancelPClub = 1 Then
        MsgBox "The customer is not currently enrolled. Please select enroll to
            continue.", vbExclamation
        Exit Sub
    End If

    'Verify source box is selected.
    If Me.cboSource.ListIndex = -1 Then

        MsgBox "Please select a source for the enrollment.", vbCritical
        Me.cboSource.SetFocus
        Exit Sub

    End If

    'Check to see if account was once enrolled, cancelled
    'and then requested to enroll again within 3 months of the cancellation.
    If Trim(Me.txtDateCanceled) = "" Then
        dteCanceled = Date
    Else
        dteCanceled = Me.txtDateCanceled
    End If

    If DateDiff("m", dteCanceled, Date) <= 3 Then
        'This is a reenrollment with the first three months of cancellation.
        Call objPClub.ReenrollAccount(objPClub.AccountNumber,
            Me.cboSource.ItemData(Me.cboSource.ListIndex))
        Call UpdatePClubForm(Me.imgEnrolled, "Reenrollment Complete...")
        Beep
        Exit Sub
    Else
        'Add or update the PClub record.
        Call objPClub.EnrollPClub(objPClub.AccountNumber,
            Me.cboSource.ItemData(Me.cboSource.ListIndex))
        Exit Sub
    End If

End If

Case 2 'Never enrolled.

'Enroll is the only option.
If Me.chkEnrolled = 0 And Me.chkCancelPClub = 1 Then
    MsgBox "The customer is not currently enrolled. Please select enroll to
        continue.", vbExclamation
    Exit Sub
End If

'Verify source box is selected.
If Me.cboSource.ListIndex = -1 Then

```



```

        MsgBox "Please select a source for the enrollment.", vbCritical
        Me.cboSource.SetFocus
        Exit Sub
    Else
        'Add or update the PClub record.
        Call objPClub.EnrollPClub(objPClub.AccountNumber,
            Me.cboSource.ItemData(Me.cboSource.ListIndex))
        Exit Sub
    End If

Case 3 'Pending reenrollment

    MsgBox "This account is pending reenrollment. No action can be taken.", vbCritical
    Exit Sub

End Select

Exit Sub

EXIT_PROC:
    Exit Sub

ActivityErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Call objPClub.AppendError(objPClub.AccountNumber, "procActivity: " & Left(Err.Description, 75))
Resume EXIT_PROC

End Sub

```

```

-----
Sub:          VerifyCancelReason
Programmer:   Paul Patterson
Last Modified: 1/18/2001
Purpose:      Enable the appropriate command buttons.
Inputs:       The value of the current flag.
Returns:      The command buttons on the form will be enabled accordingly.
-----

```

```
Function VerifyCancelReason() As Boolean
```

```
    Dim i As Integer
```

```
    VerifyCancelReason = False
```

```
    For i = 0 To Me.lstCancel.ListCount - 1
```

```
        If Me.lstCancel.Selected(i) = True Then
```

```
            VerifyCancelReason = True
```

```
            Exit Function
```

```
        End If
```

```
    Next
```

```
End Function
```

```
Sub SaveCancelReason(ByVal CancelID As String)
```

```
On Error GoTo SaveReasonErr
```

```
    Dim i As Integer
```

```
    Dim strSql As String
```

```
    For i = 0 To Me.lstCancel.ListCount - 1
```

```
        If Me.lstCancel.Selected(i) = True Then
```

```
            strSql = "INSERT INTO tblPCLubCancelReason (fkCancelID, reasonCode, cancelDate) " & _
                & "VALUES ('" & CancelID & "', '" & Me.lstCancel.ItemData(i) & "', '" & Date & "')"

```

```
            db.Execute strSql
```

```
        End If
```

```
    Next
```

```
Exit Sub
```

```
EXIT_PROC:
```

```
Exit Sub
```

```
SaveReasonErr:
```

```
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
```

```
Resume EXIT_PROC
```

```
End Sub
```

```
-----  
' Sub:          Form_Load  
' Programmer:   Paul Patterson  
' Created on:   12/20/2000  
' Purpose:      Validate the form and begin the PClub process.  
-----
```

```
Private Sub Form_Load()
```

```
    Dim rsSrc As ADODB.Recordset  
    Dim rsCancelRsn As ADODB.Recordset  
    Dim i As Integer
```

```
    Me.Height = 4650  
    Me.Width = 8145  
    Me.Left = (frmMdiMain.Width - Me.Width) / 2  
    Me.Top = 750
```

```
    'Fill the Source combo box.
```

```
    Set rsSrc = GetSource
```

```
    Do Until rsSrc.EOF
```

```
        Me.cboSource.AddItem rsSrc!Source  
        Me.cboSource.ItemData(Me.cboSource.NewIndex) = rsSrc!pksourceid  
        rsSrc.MoveNext
```

```
    Loop
```

```
    'Fill the cancel reason list box.
```

```
    Set rsCancelRsn = GetCancelReason
```

```
    Do Until rsCancelRsn.EOF
```

```
        Me.lstCancel.AddItem rsCancelRsn!reason  
        Me.lstCancel.ItemData(Me.lstCancel.NewIndex) = rsCancelRsn!CancelID  
        rsCancelRsn.MoveNext
```

```
    Loop
```

```
    ReadyPClubForm
```

```
End Sub
```

```
'-----  
' Function:           GetSource  
' Programmer:        Paul Patterson  
' Created on:         11/30/2000  
' Purpose:           Get the list of possible sources for premium club.  
' Pre:                 
' Post:              A recordset representing all available sources.  
'-----
```

```
Private Function GetSource() As ADODB.Recordset  
On Error GoTo SourceErr  
  
    Dim rsSource As ADODB.Recordset  
    Set rsSource = New ADODB.Recordset  
  
    rsSource.Open "select * from tblPclubSourceList", db  
  
    If rsSource.RecordCount < 1 Then  
        MsgBox "No matches found."  
        Exit Function  
    End If  
  
    Set GetSource = rsSource  
  
    Exit Function  
  
SourceErr:  
    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description  
  
End Function
```

```
-----  
' Function:      GetCancelReason  
' Programmer:    Paul Patterson  
' Created on:    12/14/2000  
' Purpose:       Get the list of cancel reasons for premium club.  
' Pre:  
' Post:          A recordset representing all cancel reasons.  
-----
```

```
Private Function GetCancelReason() As ADODB.Recordset  
On Error GoTo SourceErr
```

```
    Dim rsCancelReason As ADODB.Recordset  
    Set rsCancelReason = New ADODB.Recordset  
  
    rsCancelReason.Open "select * from tblPClubCancelList", db
```

```
    If rsCancelReason.RecordCount = 0 Then  
        MsgBox "No matches found."  
        Exit Function  
    End If
```

```
    Set GetCancelReason = rsCancelReason
```

```
    Exit Function
```

```
SourceErr:  
    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
```

```
End Function
```

```

-----
' Sub: MaskEdAcctNum_KeyPress
' Programmer: Paul Patterson
' Created on: 12/28/2000
' Purpose: Determine what key the user pressed and take
' appropriate action.
' Pre: keyascii
' Post:
-----
Private Sub MaskEdAcctNum_KeyPress(KeyAscii As Integer)

    If KeyAscii >= 48 And KeyAscii <= 57 Then

        Exit Sub

    'Backspace
    ElseIf KeyAscii = vbKeyBack Then
        Exit Sub

    'Copy or paste
    ElseIf KeyAscii = 3 Or KeyAscii = 22 Then
        Exit Sub

    ElseIf KeyAscii = vbKeyReturn Then

        'Verify that 16 digits are keyed for the account number on enter key.
        If Len(Me.MaskEdAcctNum) = 16 Then

            'Retrieve the account information.
            Set objPClub = New PClub
            objPClub.AccountNumber = Me.MaskEdAcctNum
            Call SetButtons(False)

            'Get the account information.
            If objPClub.GetPClubAccountInfo(objPClub.AccountNumber) = True Then
                Call SetButtons(True)
            End If

        Else

            MsgBox "The account number must be 16 digits in length, you have keyed " &
                Len(Me.MaskEdAcctNum) & ".", vbExclamation
            Exit Sub

        End If

    Else

        MsgBox "The account number can only contain numbers."
        KeyAscii = 0
        Me.MaskEdAcctNum.SetFocus

    End If

End Sub

```

Roborep System Source Code:

Description: This is the source code added to the Roborep system to accommodate the Premium Club program.

```
'-----
' Sub:                PClubCreateLabels
' Programmer:         Paul Patterson
' Created on:          11/24/2000
' Purpose:             Write the Premium Labels to a csv file. This file is
'                     emailed to
'                     qualified presort.
' Pre:                Accounts that requested to be signed up in the Premium
'                     Club.
' Post:               A csv file containing the accounts.
'-----
```

```
' Modification History
' 03/16/2001 (PP)      Modified sub to print labels from dbFDR instead of
'                     the PClub access database.
'-----
```

```
Sub PClubCreateLabels()
```

```
On Error GoTo PClubCreateLabelsErr
```

```
Dim strFileName As String
Dim strFname As String
Dim strLname As String
Dim strAddress As String
Dim strCity As String
Dim strState As String
Dim strZip As String
Dim strSql As String
Dim rsLabels As ADODB.Recordset
Dim db As ADODB.Connection
```

```
Set rsLabels = New ADODB.Recordset
```

```
strSql = "select * " _
& "From " _
& "tblPclubDetail inner join tblPclub on tblPclubDetail.accountNumber = " _
tblPclub.accountNumber " _
& "Where " _
& "labelPrintDate is null and enrollFlag = '1'"
```

```
rsLabels.Open strSql, cn, adOpenStatic, adLockOptimistic
```

```
'If no records then exit process.
If rsLabels.RecordCount = 0 Then
```

```
MsgBox "No accounts to process.", vbInformation, "Premium Club"
frmMenu.lstActivity.AddItem Now & "    No premium club accounts to process."
Exit Sub
```

```
End If
```

```
strFileName = App.Path & "\Logs\PClub\Labels\PClubLabels_" & Format(Now, "mmddyy_hhmmss")
& ".csv"
```

```
Open strFileName For Append As #1
```

```
'Process accounts.
Do Until rsLabels.EOF
```

```
'Convert to labels
strFname = StrConv(Trim(Right(rsLabels![name], (Len(rsLabels![name]) -
InStr(rsLabels![name], ","))), 3)
```

```
If InStr(rsLabels![name], ",") = 0 Then
```

```
strLname = ""
```

```

Else
    strLName = StrConv(Trim(Left(rsLabels![name], InStr(rsLabels![name], ",") - 1)), 3)
End If

strAddress = Left(StrConv(rsLabels![address], 3), 30)
strCity = StrConv(Trim(rsLabels![City]), 3)
strState = rsLabels![State]
strZip = rsLabels![Zip]

'Print to file.
Print #1, strFname & " " & strLName & "," & strAddress & "," & strCity & "," & strState &
"," & strZip

'Update the record in the database.
cn.Execute "UPDATE tblPclubDetail SET labelPrintDate = '" & Date & "' WHERE pkDetailID =
" & rsLabels!pkDetailID

'Add a note to the account.
'Call AddNote(rsLabels![tblPclub].[ASN], 3, "Pclub packet was mailed")

rsLabels.MoveNext

Loop

Close #1

'Send email notification.
With objMail

    .Recipient = "lists@qualifiedpresort.com"
    .Subject = "Attention: Brian Shaw, Premium Club Packets"
    .Body = "Here are the current labels for Premium Club."
    .Attachment = strFileName
    .SendMail

End With

frmMenu.lstActivity.AddItem Now & "    Premium club labels have been created."

EXIT_PROC:
Exit Sub

PclubCreateLabelsErr:
Call ErrorHandler
Resume EXIT_PROC:

End Sub

```



```

'-----
' Sub:          PClubDeleteProcess
' Created on:   1/15/2001
' Programmer:   Paul Patterson
' Purpose:      Process to cancel PClub enrollment.
' PRE:          A valid Pclub account.
' POST:
'-----

```

```

Public Function PClubDeleteProcess() As Boolean
On Error GoTo PClubDeleteProcessError

```

```

Dim strmsg As String
Dim intRecord As Integer
Dim intErrorCount As Integer
Dim intRow As Integer
Dim strPClubLogFile As String
Dim rsPClub As New ADODB.Recordset
Dim db As New ADODB.Connection

```

```

PClubDeleteProcess = False

```

```

'Get the location of the pclub database.

```

```

With frmMenu.dlgCommonDialog

```

```

    .FileName = "*.mdb"

```

```

    .Filter = "*.mdb"

```

```

    .ShowOpen

```

```

    If .FileName = "" Or .FileName = "*.mdb" Then Exit Function

```

```

End With

```

```

'Open the database.

```

```

    db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data Source=" &
    frmMenu.dlgCommonDialog.FileName & ";"

```

```

'Retrieve only the account numbers that haven't been enrolled.

```

```

Set rsPClub = GetPClubDeleteData(db)

```

```

'Test to verify there are account numbers to process.

```

```

If rsPClub.RecordCount <= 0 Then

```

```

    Set rsPClub = Nothing

```

```

    MsgBox "There are no account numbers to process. Please check the database.", vbOKOnly +
    vbInformation, "Notice"

```

```

    Exit Function

```

```

End If

```

```

'Create header for PClub log file.

```

```

    strPClubLogFile = "x:\applications development\logs\PClub\UnEnroll\PClubDel" &
    Format(Now, "mmddyy_hhmmss") & ".txt"

```

```

strmsg = ""

```

```

strmsg = strmsg & "Results for Premium Club deletion." & vbCrLf

```

```

strmsg = strmsg & vbCrLf

```

```

strmsg = strmsg & "Started upload on: " & Now & vbCrLf

```

```

strmsg = strmsg & vbCrLf

```

```

strmsg = strmsg & "Errd Account Numbers: " & vbCrLf

```

```

strmsg = strmsg & vbCrLf

```

```

strmsg = strmsg & "Account Number      Source      Error Message"

```

```

strmsg = strmsg & "-----"

```

```

Call AppendToFile(strmsg, strPClubLogFile)

```

```

intRecord = 0

```

```

intErrorCount = 0

```

```

frmMenu.pbr1.Min = 0

```

```

frmMenu.pbr1.Max = rsPClub.RecordCount

```

```

'Deletes customer from the Premium Club program.

```

```

Do While Not rsPClub.EOF

```

```

    intRecord = intRecord + 1

```

```

    'Status form.

```

```

    frmMenu.pbr1.Value = intRecord

```

```
frmMenu.StatusBar1.Panels(3) = "Processing record " & intRecord & " of " &
rsPClub.RecordCount
```

```
'Keystrokes to enroll customer.
objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "FPX " & rsPClub!AcctNumber
objFDR.FDRSendKeys "<EraseEOF>"
objFDR.FDRSendKeys "<Enter>"
If objFDR.FDRWatchFPX(TIMEOUT) = 0 Then
```

```
LSRetry:
```

```
'Find which row is the PClub fee and delete it.
intRow = 11 'Starting row.
```

```
strmsg = objFDR.FDRGetString(intRow, 15, 8)
If Trim(strmsg) <> "" Then
Do
```

```
'Read a line of data.
strmsg = Trim(objFDR.FDRGetString(intRow, 15, 8))
```

```
If strmsg = "PCLUB395" Or strmsg = "SECPCLUB" Or strmsg = "FSBCLUB" _
Or strmsg = "CLUBFNC" Or strmsg = "CLUBSSB" Then
```

```
'If program already deleted update the record.
If Trim(objFDR.FDRGetString(intRow, 6, 8)) <> "" Then
```

```
Call UpdatePClubAccount(db, rsPClub!acctnumberid)
Exit Do
```

```
End If
```

```
objFDR.FDRMoveTo intRow, 3
objFDR.FDRSendKeys "D"
objFDR.FDRWaitCommand "<PF2>", 10
```

```
'checks to see if this acct num is closed and another exists
```

```
If objFDR.FDRGetString(24, 2, 15) = "ACTION CODE REQ" Then
objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "BS"
objFDR.FDRWaitCommand "<Enter>", 10
If objFDR.FDRGetString(21, 21, 16) <> "0000000000000000" Then
objFDR.FDRSendKeys "<Home>fpx " & objFDR.FDRGetString(21, 21, 16)
objFDR.FDRWaitCommand "<Enter>", 10
GoTo LSRetry
```

```
End If
```

```
End If
```

```
If objFDR.FDRWatchNMCSVerifyDelete(TIMEOUT) = 0 Then
```

```
objFDR.FDRSendKeys "<PF11>"
```

```
If objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0 Then
```

```
'Watch for successful screen navigation...
```

```
'Switch to NM CS screen and update.
```

```
objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "NM CS"
objFDR.FDRSendKeys "<Enter>"
```

```
If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then
```

```
objFDR.FDRMoveTo 10, 50
objFDR.FDRSendKeys " "
objFDR.FDRSendKeys "<Enter>"
```

```
If objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0 Then
```

```
'Update the processed flag for this account.
Call UpdatePClubAccount(db, rsPClub!acctnumberid)
```

```

Exit Do

Else

    intErrorCount = intErrorCount
    strmsg = rsPClub!AcctNumber & "      " &
        objFDR.FDRGetString(24, 1, 75)
    Call AppendToFile(strmsg, strPClubLogFile)
    Exit Do

End If 'objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) <> 0

Else

    intErrorCount = intErrorCount
    strmsg = rsPClub!AcctNumber & "      " & objFDR.FDRGetString(24,
        1, 75)
    Call AppendToFile(strmsg, strPClubLogFile)
    Exit Do

End If 'objFDR.FDRWatchScreen("CUSTOMER SERVICE", 2, 27, TIMEOUT) = 0

Else

    intErrorCount = intErrorCount + 1
    strmsg = rsPClub!AcctNumber & "      " & objFDR.FDRGetString(24, 1, 75)
    Call AppendToFile(strmsg, strPClubLogFile)
    Exit Do

End If 'objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0

Else

    intErrorCount = intErrorCount + 1
    strmsg = rsPClub!AcctNumber & "      " & objFDR.FDRGetString(24, 1, 75)
    Call AppendToFile(strmsg, strPClubLogFile)
    Exit Do

End If 'objFDR.FDRWatchNMCSVerifyDelete(TIMEOUT) = 0

Else

    intRow = intRow + 2

End If

Loop Until strmsg = ""

End If 'null test

'Check to see if we need to page down for the rest of the display.
If intRow > 20 Then
    objFDR.FDRSendKeys "<PF8>" 'Page forward.
    objFDR.FDRWaitHostQuiet 1000 'Pause. This will very rarely, if ever, be hit.
    intRow = 11 'Start over.
Else
    'Next line...
    intRow = intRow + 1
End If

Else

    intErrorCount = intErrorCount + 1
    strmsg = rsPClub!AcctNumber & "      " & objFDR.FDRGetString(24, 1, 75)
    Call AppendToFile(strmsg, strPClubLogFile)

End If 'objFDR.FDRWatchFPX(TIMEOUT) = 0

'Get the next record.
rsPClub.MoveNext

```

```

Loop 'While Not rsPClub.EOF

'Write trailer information for PClub log file.
strmsg = ""
strmsg = strmsg & "Upload finished on : " & Now
strmsg = strmsg & vbCrLf
strmsg = strmsg & "Records Processed: " & rsPClub.RecordCount & vbCrLf
strmsg = strmsg & "Total Errors: " & intErrorCount & vbCrLf
strmsg = strmsg & vbCrLf
strmsg = strmsg & "Finished Premium Club deletion process."
Call AppendToFile(strmsg, strPClubLogFile)

PClubDeleteProcess = True

db.Close
objFDR.FDRSendKeys "<PF3>"
Exit Function

EXIT_PROC:

Set db = Nothing
Exit Function

PClubDeleteProcessError:
PClubDeleteProcess = False
Call ErrorHandler
Resume EXIT_PROC

End Function

```

```

-----
Sub: PClubEnrollAndCaptureProcess
Programmer: Paul Patterson
Created on: 01/15/2001
Purpose: Enroll customers in Premium Club and captures name and address
from screen to print labels.
PRE:
POST:
-----

```

```

-----
Modification History
03/16/2001 (PP) Modified the Function to be used for the enrollment of
accounts from the IVR only.
-----

```

```

Public Function PClubEnrollAndCaptureProcess(ByVal Source As String) As Boolean
On Error GoTo PClubEnrollAndCaptureProcessError

```

```

Dim strPClubLogFile As String
Dim strmsg As String
Dim strError As String
Dim intRow As Integer
Dim intRecord As Integer
Dim intErrorCount As Integer
Dim intSource As Integer
Dim rsPClubEnrollment As New ADODB.Recordset
Dim db As New ADODB.Connection
Dim lngCC As Long
Dim strAcc As String

```

```

PClubEnrollAndCaptureProcess = False

```

```

'Retrieve only the account numbers that haven't been enrolled.

```

```

If Source = "IVR" Then
    Set rsPClubEnrollment = GetIVRFile("Premium Club")
    intSource = 7
ElseIf Source = "CSI" Then
    Set rsPClubEnrollment = GetPClubDataCSI
    intSource = 4
End If

```

```

rsPClubEnrollment.MoveFirst

```

```

'Test to verify there are account numbers to process.

```

```

If rsPClubEnrollment.RecordCount <= 0 Then
    Set rsPClubEnrollment = Nothing
    MsgBox "There are no account numbers to process. Please check the Premium Club
database.", vbOKOnly + vbInformation, "Notice"
    Exit Function
End If

```

```

'Setup progressbar on form frmStatus.

```

```

frmMenu.pbr1.Min = 0
frmMenu.pbr1.Max = rsPClubEnrollment.RecordCount
frmMenu.pbr1.Value = 0

```

```

'Create header for PClub log file.

```

```

strPClubLogFile = App.Path & "\logs\PClub\Enroll\PClub_" & Source & "_" & Format(Now,
"mmddyy_hhmmss") & ".txt"
strmsg = ""
strmsg = strmsg & "Results for Premium Club enrollment from IVR." & vbCrLf
strmsg = strmsg & vbCrLf
strmsg = strmsg & "Started upload on: " & Now & vbCrLf
strmsg = strmsg & vbCrLf
strmsg = strmsg & "Errd Account Numbers: " & vbCrLf
strmsg = strmsg & vbCrLf
strmsg = strmsg & "Account Number          Source          Error Message" & vbCrLf
strmsg = strmsg & "-----" & vbCrLf
Call AppendToFile(strmsg, strPClubLogFile)

```

```

intRecord = 0
intErrorCount = 0

```

```

'Enroll customers in Premium Club and extract address information.
Do While Not rsPclubEnrollment.EOF

    intRecord = intRecord + 1

    frmMenu.pb1.Value = intRecord
    frmMenu.StatusBar1.Panels(3) = "Processing record " & intRecord & " of " &
    rsPclubEnrollment.RecordCount

    lngCC = IsCreditCard(rsPclubEnrollment!AccountNumber)
    strAcc = rsPclubEnrollment!AccountNumber

    If lngCC = -1 Then
        GoTo rsMove
    ElseIf lngCC <> 0 Then
        strAcc = Left(strAcc, Len(strAcc) - 1) & lngCC
    End If

    'Keystrokes to enroll customer.
    objFDR.FDRSendKeys "<Home>"
    objFDR.FDRSendKeys "FPX " & strAcc
    objFDR.FDRSendKeys "<EraseEOF><Enter>"

    If objFDR.FDRWatchFPX(TIMEOUT) = 0 Then
ReplacmentCard:
        'Enter loop to determine first open row.
        intRow = 11
        Do

            If Trim(objFDR.FDRGetString(intRow, 15, 1)) = "" Then 'Blank line.

                objFDR.FDRMoveTo intRow, 3
                objFDR.FDRSendKeys "A"

                'Test account number to see which PClub Method to use.
                'The first 6 numbers of the account number determine the method.
                Select Case Left(strAcc, 6)
                    Case "440363", "423980"
                        objFDR.FDRSendKeys "CLUBFNC"
                    Case "442722"
                        objFDR.FDRSendKeys "CLUBSSB"
                    Case "543360"
                        objFDR.FDRSendKeys "FSBCLUB"
                    Case Else
                        Exit Do
                End Select

                objFDR.FDRSendKeys "<PF2>"

                If objFDR.FDRWatchFPXUpdateSuccessful(10) = 0 Then

                    'Switch to NM CS screen and update.
                    objFDR.FDRSendKeys "<Home>"
                    objFDR.FDRSendKeys "NM CS"
                    objFDR.FDRSendKeys "<EraseEOF><Enter>"
                    If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then

                        objFDR.FDRMoveTo 10, 50
                        objFDR.FDRSendKeys "P"
                        objFDR.FDRSendKeys "<Enter>"

                        'Need to add this function to FDRFunctions...
                        If objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0 Then

                            'Capture the address of the cardmember.
                            objFDR.FDRSendKeys ("NM NA<Enter>")

                            'If successful navigation = 0 Then...
                            If objFDR.FDRWatchNMNAUpdateSuccessful(TIMEOUT) = 0 Then

```

```

        'Write captured account information to the database and
        update the PClub account.
        Call UpdatePClub(objFDR.FDRGetString(5, 28, 25),
        Trim(objFDR.FDRGetString(10, 17, 25)) & " " &
        Trim(objFDR.FDRGetString(10, 53, 25)),
        objFDR.FDRGetString(11, 17, 20), objFDR.FDRGetString(11,
        45, 2), objFDR.FDRGetString(11, 60, 9), strAcc)
        cn.Execute "UPDATE tblPClub SET enrollFlag = '1' WHERE
        accountNumber = '" & strAcc & "'"
        cn.Execute "INSERT INTO tblPClubDetail
        (accountNumber,fkSourceID,userID,enrollDate) VALUES('" &
        strAcc & "','" & intSource & "','" & Source & "','" & Date &
        "'"")
        strmsg = strAcc & "      Enroll Success!"
        Call AppendToFile(strmsg, strPClubLogFile)
        Exit Do

    Else
        intErrorCount = intErrorCount + 1
        strmsg = strAcc & "      Error on NMNA update"
        Call AppendToFile(strmsg, strPClubLogFile)
    End If

Else
    intErrorCount = intErrorCount + 1
    strmsg = strAcc & "      Error on NMCS update"
    Call AppendToFile(strmsg, strPClubLogFile)
End If 'objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0
Else
    intErrorCount = intErrorCount + 1
    strmsg = strAcc & "      Error moving to NMCS"
    Call AppendToFile(strmsg, strPClubLogFile)
End If 'If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then
Else
    'Used to move to new account for lost/stolen cards CAH 12/20/00
    If objFDR.FDRGetString(24, 2, 15) = "ACTION CODE REQ" Then
        objFDR.FDRSendKeys "<Home>cbx<Enter>"
        objFDR.FDRWaitHostQuiet (2000)
        strmsg = objFDR.FDRGetString(1, 5, 16)
        objFDR.FDRSendKeys "fpx " & strmsg & "<Enter>"
        objFDR.FDRWaitHostQuiet (1000)
        strmsg = strAcc & "      Lost/Stolen Action Code: " & strmsg
        Call AppendToFile(strmsg, strPClubLogFile)
        GoTo ReplacmentCard
    End If
    intErrorCount = intErrorCount + 1
    strmsg = strAcc & "      Error on FPX update"
    Call AppendToFile(strmsg, strPClubLogFile)
    Exit Do

    End If 'objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0

ElseIf objFDR.FDRGetString(intRow, 15, 8) = "PCLUB395" Or _
    objFDR.FDRGetString(intRow, 15, 8) = "SECPCLUB" Or _
    Trim(objFDR.FDRGetString(intRow, 15, 8)) = "CLUBSSB" Or _
    Trim(objFDR.FDRGetString(intRow, 15, 8)) = "CLUBFNC" Or _
    Trim(objFDR.FDRGetString(intRow, 15, 8)) = "FSBCLUB" Then 'Already enrolled.

    strmsg = strAcc & " Already Enrolled"
    Call AppendToFile(strmsg, strPClubLogFile)
    intErrorCount = intErrorCount + 1
    Exit Do

End If

intRow = intRow + 2

Loop 'To determine open FPX row...

Else
    intErrorCount = intErrorCount + 1

```

```

        strmsg = strAcc & "      " & objFDR.FDRGetString(24, 1, 80)
        Call AppendToFile(strmsg, strPClubLogFile)
    End If 'objFDR.FDRWatchFPX(TIMEOUT) = 0
rsMove:
    rsPClubEnrollment.MoveNext

    Loop 'Main loop...

    'Write trailer information for PClub log file.
    strmsg = ""
    strmsg = strmsg & vbCrLf
    strmsg = strmsg & "Upload finished on :      " & Now & vbCrLf
    strmsg = strmsg & vbCrLf
    strmsg = strmsg & "Records Processed:  " & rsPClubEnrollment.RecordCount & vbCrLf
    strmsg = strmsg & "Total Errors:      " & intErrorCount & vbCrLf
    strmsg = strmsg & vbCrLf
    strmsg = strmsg & "Finished Premium Club enrollment process."
    Call AppendToFile(strmsg, strPClubLogFile)

    PClubEnrollAndCaptureProcess = True

    objFDR.FDRSendKeys "<Home>"
    objFDR.FDRSendKeys "<PF3>"
    Exit Function

EXIT_PROC:
    Exit Function

PClubEnrollAndCaptureProcessError:
    PClubEnrollAndCaptureProcess = False
    Call ErrorHandler
    Resume EXIT_PROC
End Function

```



```

-----
' Sub:                PClubEnrollFromApps
' Programmer:         Paul Patterson
' Created on:         11/21/2000
' Purpose:            Process to enroll customers who choose to enroll in PClub
'                    at the time they apply for the card.
' Pre:                Accounts from dbFDR
' Post:               Log file containing the information to print labels.
-----

```

```

Sub PClubEnrollFromApps()
On Error GoTo PClubEnrollFromAppsErr

```

```

Dim intRecord As Integer
Dim intErrorCount As Integer
Dim intRow As Integer
Dim strmsg As String
Dim strPClubLogFile As String
Dim strFileName As String
Dim rsPClub As ADODB.Recordset
Dim db As ADODB.Connection

```

```

'Connection for the PClub access database.
Set db = New ADODB.Connection

```

```

'Get the location of the pclub database.
With frmMenu.dlgCommonDialog
.FileName = "*.mdb"
.Filter = "*.mdb"
.ShowOpen
If .FileName = "" Or .FileName = "*.mdb" Then Exit Sub
End With

```

```

'Open the database.
db.Open "PROVIDER=Microsoft.Jet.OLEDB.3.51;Data Source=" & frmMenu.dlgCommonDialog.FileName
& ";"

```

```

'Path and filename for the log file.
strFileName = App.Path & "\logs\PClub\PClubFromApps_" & Format(Now, "mmddyy_hhmmss") & ".txt"

```

```

Set rsPClub = New ADODB.Recordset
rsPClub.Open "Select tblA20queue.* from tblA20queue inner join tblapplications on
tblA20queue.asn = " _
& "tblapplications.asn and tblapplications.pclub = '1'", cn, adOpenStatic, adLockOptimistic

```

```

Open strFileName For Append As #1

```

```

Print #1, String(75, "**")
Print #1, "Results for PClub processing from applications"
Print #1, strFileName
Print #1, String(75, "**")
Print #1, ""
Print #1, "Started PClub processing at: " & Now
Print #1, "Records to process: " & rsPClub.RecordCount
Print #1, ""

```

```

'No records, then exit.
If rsPClub.RecordCount = 0 Then
MsgBox "No records to process.", vbInformation, "Premium Club"
Print #1, "No records to process."
Close #1
Set rsPClub = Nothing
Set db = Nothing
Exit Sub
End If

```

```

'Setup progress bar.
frmMenu.pb1.Min = 0
frmMenu.pb1.Max = rsPClub.RecordCount

```

```

'Enroll customers in Premium Club and extract address information.
Do While Not rsPClub.EOF

```

```

intRecord = intRecord + 1

frmMenu.pb1.Value = intRecord
frmMenu.StatusBar1.Panels(3) = "Processing record " & intRecord & " of " &
rsPClub.RecordCount

'Keystrokes to enroll customer.
objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "FPX " & rsPClub!AccountNumber
objFDR.FDRSendKeys "<EraseEOF><Enter>"

If objFDR.FDRWatchFPX(TIMEOUT) = 0 Then

    'Enter loop to determine first open row.
    intRow = 11
    Do

        If Trim(objFDR.FDRGetString(intRow, 15, 1)) = "" Then 'Blank line.

            objFDR.FDRMoveTo intRow, 3

            objFDR.FDRSendKeys "A"
            'Test account number to see which PClub Method to use.
            'The first 6 numbers of the account number determine the method.
            Select Case Left(rsPClub!AccountNumber, 6)
                Case "440363", "423980"
                    objFDR.FDRSendKeys "CLUBFNC"
                Case "442722"
                    objFDR.FDRSendKeys "CLUBSSB"
                Case "543360"
                    objFDR.FDRSendKeys "FSBCLUB"
                Case Else
                    Exit Do
            End Select

            objFDR.FDRSendKeys "<PF2>"

            If objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0 Then

                'Switch to NM CS screen and update.
                objFDR.FDRSendKeys "<Home>"
                objFDR.FDRSendKeys "NM CS"
                objFDR.FDRSendKeys "<EraseEOF><Enter>"
                If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then

                    objFDR.FDRMoveTo 10, 50
                    objFDR.FDRSendKeys "P"
                    objFDR.FDRSendKeys "<Enter>"

                    If objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0 Then

                        'Capture the address of the cardmember.
                        objFDR.FDRSendKeys ("NM NA<Enter>")

                        'If successful navigation = 0 Then...
                        If objFDR.FDRWatchNMNAUpdateSuccessful(TIMEOUT) = 0 Then

                            'Write captured account information to the database.
                            Call InsertIntoPClubAddressTable(db, objFDR.FDRGetString(5,
                                28, 25), Trim(objFDR.FDRGetString(10, 17, 25)) & " " &
                                Trim(objFDR.FDRGetString(10, 53, 25)),
                                objFDR.FDRGetString(11, 17, 20), objFDR.FDRGetString(11,
                                    45, 2), objFDR.FDRGetString(11, 60, 9),
                                objFDR.FDRGetString(3, 10, 16))
                            Print #1, rsPClub!AccountNumber & "      Update Successful"
                            Exit Do

                        Else

                            intErrorCount = intErrorCount + 1

```

```

        Print #1, rsPClub!AccountNumber; "      Error on NMNA update"
        'Call InsertIntoPClubErrors(cn, rsPClub!accountNumber)
    End If
Else
    intErrorCount = intErrorCount + 1
    Print #1, rsPClub!AccountNumber; "      Error on NMCS update"
End If 'Unsuccessful...

Else
    intErrorCount = intErrorCount + 1
    strmsg = rsPClub!AccountNumber & "      Error on NMCS update"
    Print #1, strmsg
    Exit Do
End If 'objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0

Else

    intErrorCount = intErrorCount + 1
    strmsg = rsPClub!AccountNumber & "      Error on FPX update"
    Call AppendToFile(strMsg, strPClubLogFile)
    Exit Do

    End If 'objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0

ElseIf objFDR.FDRGetString(intRow, 15, 8) = "PCLUB395" Or _
    objFDR.FDRGetString(intRow, 15, 8) = "SECPCLUB" Or _
    objFDR.FDRGetString(intRow, 15, 4) = "CLUB" Or _
    objFDR.FDRGetString(intRow, 15, 7) = "FSBCLUB" Then 'Already enrolled.
    strmsg = rsPClub!AccountNumber & "      Already Enrolled"
    Print #1, strmsg
    intErrorCount = intErrorCount + 1
    Exit Do

End If

    intRow = intRow + 2

Loop 'To determine open FPX row...

Else

    intErrorCount = intErrorCount + 1

    strmsg = rsPClub!AccountNumber & "      " & objFDR.FDRGetString(24, 1, 80)
    Call AppendToFile(strmsg, strPClubLogFile)

End If 'objFDR.FDRWatchFPX(TIMEOUT) = 0

rsPClub.MoveNext

Loop 'Main loop...

Print #1, ""
Print #1, "Finished PClub processing on " & Now

EXIT_PROC:
    Close #1
    Set rsPClub = Nothing
    Set db = Nothing
    Exit Sub

PClubEnrollFromAppsErr:
    Call ErrorHandler

    Resume EXIT_PROC

End Sub

```

```

-----
Sub: UpdatePClub
Programmer: Paul Patterson
Created on: 12/27/2000
Purpose: Add or update the account information in the Pclub database.
Pre: The name and address of the current account.
Post:
-----

Sub UpdatePClub(ByVal name As String, ByVal address As String, _
    ByVal City As String, _
    ByVal State As String, _
    ByVal Zip As String, _
    ByVal AccountNumber As String)
On Error GoTo updateErr

    Dim strSql As String

    'Update the PClub table with the address information.
    If CheckAccountNumber(AccountNumber) Then

        'Account number exists, update the account information.
        strSql = "UPDATE tblPClub set name = '" & name & "'," _
            & "address = '" & address & "'," _
            & "city = '" & City & "'," _
            & "state = '" & State & "'," _
            & "zip = '" & Zip _
            & "' WHERE accountNumber = '" & AccountNumber & "'"

        cn.Execute strSql

    Else

        'New account number, write the account information to the database.
        strSql = "INSERT INTO tblPClub (accountNumber, name, address, city, state, zip) " _
            & "VALUES('" & AccountNumber & "'," _
            & name & "'," _
            & address & "'," _
            & City & "'," _
            & State & "'," _
            & Zip & "')"

        cn.Execute strSql

    End If

EXIT_PROC:
Exit Sub

updateErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Call AppendError(AccountNumber, Left(Err.Description, 75))
Resume EXIT_PROC

End Sub

```

```

-----
' Sub:                GetPClubDataCSI
' Programmer:         Paul Patterson
' Created on:         03/16/2001
' Purpose:            Load the Pclub file from Central States Indemnity.
' Pre:
' Post:
-----

```

```

Public Function GetPClubDataCSI() As ADODB.Recordset
On Error GoTo FlatFileErr

    Dim strRecord As String
    Dim strAcctNumber As String
    Dim rsIVR As ADODB.Recordset

    'Get the location of the file.
    With frmMenu.dlgCommonDialog
        .FileName = "*.txt"
        .Filter = "*.txt"
        .ShowOpen
        If .FileName = "" Or .FileName = "*.txt" Then Exit Function
    End With

    Set rsIVR = New ADODB.Recordset

    'Create a field for the account number.
    With rsIVR

        .Fields.Append "AccountNumber", adVarChar, 16, adFldUpdatable
        .CursorType = adOpenKeyset
        .LockType = adLockOptimistic
        .Open

    End With

    'Open file.
    Open frmMenu.dlgCommonDialog.FileName For Input As #1

    'Loop until end of file.
    Do While Not EOF(1)

        Line Input #1, strRecord 'Read entire line into variable.

        'Get the account number.
        strAcctNumber = Left(strRecord, 16)

        'Add the account number to the recordset.
        With rsIVR
            .AddNew
            rsIVR!AccountNumber = strAcctNumber
            .Update
        End With

    Loop

    Close #1

    Set GetPClubDataCSI = rsIVR

EXIT_PROC:
    Exit Function

FlatFileErr:
    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
    Resume EXIT_PROC

End Function

```

```

-----
' Function:          CheckAccountNumber
' Programmer:       Paul Patterson
' Created on:       12/19/2000
' Purpose:         Determine if the account number already exists in the CAPS database.
' Pre:             A string representing the account number.
' Post:            True if the account exists.
-----

```

```

Public Function CheckAccountNumber(ByVal AcctNumber As String) As Boolean
On Error GoTo checkErr

```

```

    CheckAccountNumber = False

```

```

    Dim rsAcctNumber As ADODB.Recordset

```

```

    Set rsAcctNumber = New ADODB.Recordset

```

```

    'Look for an existing ASN.

```

```

    rsAcctNumber.Open "Execute sp_CheckExistingPClubAcct @AcctNumber = '" & AcctNumber & "'", cn
    If rsAcctNumber.RecordCount > 0 Then

```

```

        CheckAccountNumber = True

```

```

    End If

```

```

EXIT_PROC:

```

```

    Exit Function

```

```

checkErr:

```

```

    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description

```

```

    Resume EXIT_PROC

```

```

End Function

```

Caps System source code:

```
'-----  
' Class:      PClub (PClub.cls)  
' Programmer: Paul Patterson  
' Purpose:    Wrapper class for general PClub routines.  
' Notes:  
'-----
```

Option Explicit

```
'Member variables.  
Private m_blnEnrolled As Boolean      'Is the customer enrolled in PClub?  
Private m_strAccountNumber As String 'Current account number being worked.  
Private m_bytType As PClubTypes
```

```
Public Enum PClubTypes  
    btEnrolled = 0  
    btCancelled  
    btNeverEnrolled  
    btReenrollment  
End Enum
```

```

-----
Sub:                CheckReenrollment
Programmer:         Paul Patterson
Created on:         01/12/2001
Purpose:            Determine if an account is pending a reenrollment and update
                    the PClub form accordingly.
Pre:
Post:
-----

```

```

Function CheckReenrollment() As Boolean
On Error GoTo reenrollErr

```

```

    CheckReenrollment = False

```

```

    Dim strSql As String
    Dim rsReenroll As ADODB.Recordset
    Set rsReenroll = New ADODB.Recordset

```

```

    strSql = "SELECT requestDate from tblPClubReenroll where accountNumber = '" & AccountNumber
    & "' and reenrollDate is null"

```

```

    rsReenroll.Open strSql, db

```

```

    If rsReenroll.RecordCount > 0 Then

```

```

        CheckReenrollment = True
        PClubType = btReenrollment
        Call frmPClub.UpdatePClubForm(frmPClub.imgNotEnrolled, "This account is pending a
        reenrollment requested on " & rsReenroll!requestdate & ".")

```

```

        rsReenroll.Close
        Set rsReenroll = Nothing

```

```

    End If

```

```

    Exit Function

```

```

EXIT_PROC:
    Exit Function

```

```

reenrollErr:
    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
    Resume EXIT_PROC

```

```

End Function

```



```

Public Property Get PClubType() As PClubTypes

    PClubType = m_bytType

End Property

Public Property Let PClubType(ByVal NewType As PClubTypes)

    Select Case NewType
        Case btEnrolled, btCancelled, btNeverEnrolled, btReenrollment
            ' No need to do anything if NewType is valid.
        Case Else
            Err.Raise Number:=vbObjectError + 32112, _
                Description:="Invalid PClub type"
    End Select

    m_bytType = NewType

End Property

Public Property Get AccountNumber() As String

    AccountNumber = m_strAccountNumber

End Property

Public Property Get EnrollmentStatus() As Boolean

    EnrollmentStatus = m_blnEnrolled

End Property

Public Property Let AccountNumber(ByVal AcctNumber As String)

    m_strAccountNumber = AcctNumber

End Property

Public Property Let EnrollmentStatus(ByVal Status As Boolean)

    m_blnEnrolled = Status

End Property

```

```

-----
Sub:                ReenrollAccount
Programmer:         Paul Patterson
Created on:         12/27/2000
Purpose:            If the account is reenrolling after canceling, within
                    3 months of
                    cancellation then log for report.
Pre:                A string representing the account number.
Post:
-----

```

```

Sub ReenrollAccount(ByVal AcctNumber As String, ByVal Source As String)
On Error GoTo reenrollErr

```

```

    db.Execute ("INSERT INTO tblPClubReenroll (accountNumber, requestDate, fkSourceID, userID) "
    & "Values('" & AcctNumber & "','" & Date & "','" & Source & "','" & userID & "'))"

```

```

Exit Sub

```

```

EXIT_PROC:
Exit Sub

```

```

reenrollErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Call AppendError(AcctNumber, "Error inserting reenrollment record")
Resume EXIT_PROC

```

```

End Sub

```

```

'-----
' Sub: AppendError
' Programmer: Paul Patterson
' Last Modified: 12/19/2000
' Purpose: An error message is written to the log file.
' PRE: A string representing the error message.
' POST:
'-----

```

```

Public Sub AppendError(ByVal AcctNumber As String, ByVal Message As String)
On Error GoTo AppendErrorError

```

```

    db.Execute "INSERT INTO tblPClubError (accountNumber,errTime,userID, errMessage) VALUES('" &
    AcctNumber _
    & "','" & Now() & "','" & userID & "','" & Message & "'"")"

```

```

    Beep
    Call frmPClub.ReadyPClubForm
    Call frmPClub.UpdatePClubForm(frmPClub.imgError, Message)
    Screen.MousePointer = vbDefault
    Exit Sub

```

```

EXIT_PROC:
    Exit Sub

```

```

AppendErrorError:
    MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
    Resume EXIT_PROC
End Sub

```

```

-----
Function:      GetPClubAccountInfo
Programmer:    Paul Patterson
Created on:    11/30/2000
Purpose:       Verify a valid account number in FDR.
Pre:           A string representing the account number.
Post:         False on failure
-----

```

```

Public Function GetPClubAccountInfo(ByVal acctNum As String) As Boolean
On Error GoTo verifyErr

```

```

Dim strMsg As String           'The message displayed on the PClub form.
Dim strBSAddress As String
Dim strName As String
Dim strAddress As String
Dim strCity As String
Dim strState As String
Dim strZip As String
Dim strDateEnrolled As String
Dim strLastBillingDate As String
Dim strNextBillingDate As String
Dim strDateCanceled As String
Dim strOpenDate As String
Dim strCLIDate As String
Dim intX As Integer

```

```

GetPClubAccountInfo = False

```

```

'Prepare the PClub form.
frmPClub.ReadyPClubForm
Screen.MousePointer = vbHourglass
Call frmPClub.UpdatePClubForm(frmPClub.imgFDR, "Please wait while FDR connection is
established...")

```

```

'Run a mod 10 check on the account to validate.
If MOD10Check(acctNum) Then

```

```

    'Verify the connection status.
    If objFDR.FDRGetConnectionStatus = 0 Then

```

```

        MsgBox "Unable to sign on to the FDR system. Please close the FDR session and retry.",
vbCritical
Call frmPClub.UpdatePClubForm(frmPClub.imgError, "Error connecting to the FDR system.")
        frmPClub.MaskEdAcctNum.SetFocus
        Screen.MousePointer = vbDefault
        Exit Function

```

```

    End If

```

```

    'Capture the account information.
    Call frmPClub.UpdatePClubForm(frmPClub.imgWait, "Please wait while account information is
obtained...")
    frmMdiMain.SetFocus
    With objFDR

```

```

        'Go to the BS screen.
        .FDRSendKeys ("<PF3>")
        .FDRSendKeys ("<Home>")
        .FDRSendKeys "BS " & acctNum & "<EraseEOF>"
        .FDRSendKeys ("<Enter>")
        .FDRWaitHostQuiet
        .FDRWatchBS TIMEOUT

```

```

    If objFDR.FDRWatchBS(TIMEOUT) = 1 Then

```

```

        strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
        Call AppendError(AccountNumber, strMsg)
        GetPClubAccountInfo = False
        Exit Function

```

```

    End If

```

```

'Get both lines of the address from FDR.
strBSAddress = Trim(objFDR.FDRGetString(2, 1, 80)) & Trim(objFDR.FDRGetString(3,
1, 80))

strName = ""
strAddress = ""
strCity = ""
strState = ""
strZip = ""
intX = 0

'Parse the address string.
strName = Mid(strBSAddress, 1, InStr(1, strBSAddress, "*") - 1)
intX = Len(strName) + 2
intX = InStr(intX, strBSAddress, "*") + 1
strAddress = Mid(strBSAddress, intX, InStr(intX, strBSAddress, "*") - intX)
intX = InStr(intX, strBSAddress, "*") + 1
strAddress = strAddress & " " & Mid(strBSAddress, intX, InStr(intX, strBSAddress,
"") - intX)
intX = InStr(intX, strBSAddress, "*") + 1
strCity = Mid(strBSAddress, intX, InStr(intX, strBSAddress, "*") - intX)
intX = InStr(intX, strBSAddress, "*") + 1
strState = Mid(strBSAddress, intX, InStr(intX, strBSAddress, "*") - intX)
intX = InStr(intX, strBSAddress, "*") + 1
strZip = Mid(strBSAddress, intX, InStr(intX, strBSAddress, "*") - intX)

strOpenDate = objFDR.FDRGetString(6, 46, 5) 'Open date
strCLIDate = objFDR.FDRGetString(10, 70, 5) 'CLI date

'Go to the FPX screen.
.FDRSendKeys "<Home>"
.FDRSendKeys "FPX"
.FDRSendKeys "<Enter>"
.FDRWatchFPX TIMEOUT

'Check to see if account is enrolled.
If SearchPClub = True Then

    'Get the activity dates.
    strDateEnrolled = objFDR.FDRGetString(objFDR.FDRRow, 71, 8)
    strLastBillingDate = objFDR.FDRGetString(objFDR.FDRRow + 1, 61, 8)

    'Test next billing date for cancellation.
    If objFDR.FDRGetString(objFDR.FDRRow, 60, 8) = "99/99/99" Then

        'PClub has been canceled.
        strNextBillingDate = "Canceled"
        frmPClub.chkCancelPClub = 1
        PClubType = btCancelled

        Call frmPClub.UpdatePClubForm(frmPClub.imgNotEnrolled, "This account has
        canceled PClub.")
        Call LoadCancelReason

    Else

        'Currently enrolled in PClub.
        strNextBillingDate = objFDR.FDRGetString(objFDR.FDRRow, 60, 8)
        frmPClub.chkEnrolled = 1
        PClubType = btEnrolled
        Call LoadEnrollSource
        Call frmPClub.UpdatePClubForm(frmPClub.imgEnrolled, "This account is
        enrolled in PClub.")

    End If

    strDateCanceled = objFDR.FDRGetString(objFDR.FDRRow, 6, 8)

Else

```

```

'Never enrolled.
frmPClub.ReadyPClubForm
PClubType = btNeverEnrolled
Call frmPClub.UpdatePClubForm(frmPClub.imgNotEnrolled, "This account has not been
enrolled in PClub.")

```

End If

```

'Check for reenrollment.
Call CheckReenrollment

```

```

'Populate the form.
frmPClub.txtCLIDate = strCLIDate
frmPClub.txtOpenDate = strOpenDate
frmPClub.txtDateCanceled = strDateCanceled
frmPClub.txtDateEnrolled = strDateEnrolled
frmPClub.txtLastBillingDate = strLastBillingDate
frmPClub.txtNextBillingDate = strNextBillingDate
frmPClub.txtName = strName
frmPClub.txtAddress = strAddress
frmPClub.txtCity = strCity
frmPClub.txtState = strState
frmPClub.txtZip = strZip

```

End With

Else

```

MsgBox "This is not a valid account number. Please verify the account number.",
vbCritical
GetPClubAccountInfo = False
Call frmPClub.UpdatePClubForm(frmPClub.imgError, "Invalid account number.")
Screen.MousePointer = vbDefault
frmPClub.MaskEdAcctNum.SetFocus
frmPClub.MaskEdAcctNum.SelStart = 0
frmPClub.MaskEdAcctNum.SelLength = Len(frmPClub.MaskEdAcctNum.Text) + 3
Exit Function

```

End If

```

Call UpdatePClub(strName, strAddress, strCity, strState, strZip)
Screen.MousePointer = vbDefault
GetPClubAccountInfo = True
Exit Function

```

```

EXIT_PROC:
Screen.MousePointer = vbDefault
frmPClub.ReadyPClubForm
Exit Function

```

```

verifyErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Call AppendError(AccountNumber, Left(Err.Description, 75))
GetPClubAccountInfo = False
Resume EXIT_PROC

```

End Function

```

-----
' Sub:           UpdatePClub
' Programmer:    Paul Patterson
' Created on:    12/27/2000
' Purpose:       If the account number is already available then update the
'                information.
' Pre:
' Post:
-----

Sub UpdatePClub(ByVal name As String, _
                ByVal address As String, _
                ByVal City As String, _
                ByVal State As String, _
                ByVal Zip As String)

On Error GoTo updateErr

    Dim strSql As String

    'Update the PClub table with the address information.
    If CheckAccountNumber(AccountNumber) Then

        'Account number exists, update the account information.
        strSql = "UPDATE tblPClub set name = '" & name & "'," _
                & "address = '" & address & "'," _
                & "city = '" & City & "'," _
                & "state = '" & State & "'," _
                & "zip = '" & Zip _
                & "' WHERE accountNumber = '" & AccountNumber & "'"

        db.Execute strSql

    Else

        'New account number, write the account information to the database.
        strSql = "INSERT INTO tblPClub (accountNumber, name, address, city, state, zip) " _
                & "VALUES('" & AccountNumber & "'," _
                & name & "'," _
                & address & "'," _
                & City & "'," _
                & State & "'," _
                & Zip & "')"

        db.Execute strSql

    End If

Exit Sub

EXIT_PROC:
Exit Sub

updateErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Call AppendError(AccountNumber, Left(Err.Description, 75))
Resume EXIT_PROC

End Sub

```

```

-----
Function:      MOD10Check
Programmer:    Paul Patterson, 2/9/2000
Purpose:       Mod 10 check algorithm; validates 16 digit credit
               card numbers.
Pre:           16 character string value representing account no.
Post:          Returns TRUE if the card number is validated by
               the algorithm.
-----

```

```
Public Function MOD10Check(ByVal CardNumber$) As Boolean
```

```

    Dim i As Integer
    Dim J As Integer
    Dim iTemp As Integer
    Dim iFinal As Integer
    Dim sTemp As String

```

```
sTemp = ""
```

```
'Parse out the - in the account number text.
```

```
'XXXX-XXXX-XXXX-XXXX
```

```
For i = 1 To Len(CardNumber$)
```

```

    If Mid$(CardNumber$, i, 1) <> "-" Then
        sTemp = sTemp & Mid$(CardNumber$, i, 1)
    End If

```

```
Next i
```

```
iTemp = 0
```

```
i = 1
```

```
For J = 1 To 15 Step 2
```

```
    iTemp = (val(Mid$(sTemp, J, 1)) * 2)
```

```
    If Len(CStr(iTemp)) = 2 Then
```

```

        iTemp = val(Mid$(sTemp, i + 1, 1)) + val(Right$(CStr(iTemp), 1)) +
        val(Left$(CStr(iTemp), 1))
    End If

```

```
    ElseIf Len(CStr(iTemp)) = 1 Then
```

```
        iTemp = val(Mid$(sTemp, i + 1, 1)) + iTemp
```

```
    End If
```

```
    iFinal = iFinal + iTemp
```

```
    i = i + 2
```

```
    iTemp = 0
```

```
Next
```

```
MOD10Check = ((iFinal Mod 10) = 0)
```

```
End Function
```



```

-----
Sub:                EnrollPCLub
Programmer:         Paul Patterson
Created on:         12/19/2000
Purpose:            Enroll customers in Premium Club and captures name and address
                    from screen to print labels.
Pre:
Post:
-----

```

```

Public Function EnrollPCLub(ByVal AcctNumber As String, ByVal Source As String) As Boolean
On Error GoTo PCLubEnrollAndCaptureProcessError

```

```

Dim strPCLubLogFile As String
Dim strMsg As String
Dim strError As String
Dim strSourceCode As String
Dim intRow As Integer
Dim intRecord As Integer
Dim intErrorCount As Integer

```

```

EnrollPCLub = False
Call frmPCLub.UpdatePCLubForm(frmPCLub.imgFDR, "Please wait while FDR connection is
established...")

```

```

'Verify the connection status.

```

```

If objFDR.FDRGetConnectionStatus = 0 Then

```

```

    MsgBox "Unable to sign on to the FDR system. Please close the FDR session and
    retry.", vbCritical
    frmPCLub.lblMessage = "Unable to sign on to the FDR system."
    frmPCLub.MaskEdAcctNum.SetFocus
    Screen.MousePointer = vbDefault
    Exit Function

```

```

End If

```

```

'Enroll customer in Premium Club.

```

```

Call frmPCLub.UpdatePCLubForm(frmPCLub.imgWait, "Enrolling account in PCLub, please wait...")

```

```

objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "BS8 " & AcctNumber
objFDR.FDRSendKeys "<EraseEOF><Enter>"
If objFDR.FDRWatchBS8(TIMEOUT) = 0 Then

```

```

    strSourceCode = objFDR.FDRGetString(7, 74, 4)

```

```

Else

```

```

    strMsg = Trim(objFDR.FDRGetString(24, 1, 80))
    Call AppendError(AcctNumber, strMsg)
    Call frmPCLub.UpdatePCLubForm(frmPCLub.imgError, "Error on BS8 - " & strMsg)
    Exit Function

```

```

End If

```

```

objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "FPX " & AcctNumber
objFDR.FDRSendKeys "<EraseEOF><Enter>"

```

```

If objFDR.FDRWatchFPX(TIMEOUT) = 0 Then

```

```

    'Enter loop to determine first open row.
    intRow = 11
    Do

```

```

        If Trim(objFDR.FDRGetString(intRow, 15, 1)) = "" Then 'Blank line.

```

```

            objFDR.FDRMoveTo intRow, 3
            objFDR.FDRSendKeys "A"
            objFDR.FDRSendKeys GetPCLubMethod(strSourceCode)

```

```

objFDR.FDRSendKeys "<PF2>"

If objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0 Then

    'Switch to NM CS screen and update.
    objFDR.FDRSendKeys "<Home>"
    objFDR.FDRSendKeys "NM CS"
    objFDR.FDRSendKeys "<EraseEOF><Enter>"
    If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then

        objFDR.FDRMoveTo 10, 50
        objFDR.FDRSendKeys "P"
        objFDR.FDRSendKeys "<Enter>"

        If objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 1 Then

            strMsg = "Error on NMNA update"
            Call AppendError(AcctNumber, strMsg)

        Else

            'Update the PClub account with the enrollment date.
            db.Execute "UPDATE tblPClub SET enrollFlag = '1' WHERE
            accountNumber = '" & AcctNumber & "'"
            db.Execute "INSERT INTO tblPClubDetail
            (accountNumber,fkSourceID,userID,enrollDate) VALUES('" & AcctNumber
            & "','" & Source & "','" & userID & "','" & Date & "'"
            Exit Do

        End If

    Else

        strMsg = "Error on NMCS update"
        Call AppendError(AcctNumber, strMsg)

    End If 'objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0

Else

    strMsg = "Error on FPX update"
    Call AppendError(AcctNumber, strMsg)
    Exit Do

End If 'pp objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0

ElseIf (objFDR.FDRGetString(intRow, 15, 8) = "PCLUB395" And
Trim(objFDR.FDRGetString(intRow, 6, 8)) = "") Or _
(objFDR.FDRGetString(intRow, 15, 8) = "SECPCLUB" And
Trim(objFDR.FDRGetString(intRow, 6, 8)) = "") Then 'Already enrolled.

    strMsg = "Already Enrolled"
    Call AppendError(AcctNumber, strMsg)
    Exit Do

End If 'If Trim(objFDR.FDRGetString(intRow, 15, 1)) = "" Then 'Blank line.

intRow = intRow + 2

Loop 'To determine open FPX row...

Else

    strMsg = Trim(objFDR.FDRGetString(24, 1, 80))
    Call AppendError(AcctNumber, strMsg)

End If 'objFDR.FDRWatchFPX(TIMEOUT) = 0

EnrollPClub = True

objFDR.FDRSendKeys "<Home>"

```

```
objFDR.FDRSendKeys "<PF3>"
Call frmPClub.UpdatePClubForm(frmPClub.imgEnrolled, "Enrollment complete...")
Beep
Exit Function
```

```
EXIT_PROC:
Exit Function
```

```
PClubEnrollAndCaptureProcessError:
EnrollPClub = False
Call AppendError(AcctNumber, Left(Err.Description, 75))
Resume EXIT_PROC
```

```
End Function
```

```

-----
Function:      LoadCancelReason
Programmer:    Paul Patterson
Created on:    11/30/2000
Purpose:       Load any items that were selected in the cancel list.
Pre:
Post:
-----

```

```

Function LoadCancelReason() As Boolean
On Error GoTo LoadErr

```

```

Dim i As Integer
Dim rsCancel As ADODB.Recordset
Dim rsID As ADODB.Recordset

```

```

Set rsCancel = New ADODB.Recordset
Set rsID = New ADODB.Recordset

```

```

'Get the most current cancel reason.
rsID.Open ("SELECT MAX(pkCancelID) as [pkCancelID] from tblPclubCancel WHERE
accountNumber = '" & AccountNumber & "'", db

```

```

'No cancel reasons, exit proc.
If IsNull(rsID!pkcancelid) Then Exit Function

```

```

'At least one cancel reason exists.
rsCancel.Open ("SELECT DISTINCT reasonCode from tblPclubCancelReason WHERE fkCancelID =
'" & rsID!pkcancelid & "'", db

```

```

'Populate the cancel reason list box.
If rsCancel.RecordCount > 0 Then

```

```

    Do Until rsCancel.EOF

```

```

        frmPclub.lstCancel.Selected(rsCancel!reasonCode - 1) = True
        rsCancel.MoveNext
    
```

```

    Loop

```

```

End If

```

```

'Close the recordsets.
rsID.Close
rsCancel.Close

```

```

Exit Function

```

```

EXIT_PROC:
Exit Function

```

```

LoadErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Resume EXIT_PROC

```

```

End Function

```

```

-----
Function:      LoadEnrollSource
Programmer:    Paul Patterson
Created on:    01/05/2001
Purpose:       Load the source of the enrollment into the source combo box.
Pre:
Post:
-----

```

```

Function LoadEnrollSource() As Boolean
On Error GoTo LoadErr

```

```

Dim i As Integer
Dim rsCancel As ADODB.Recordset
Dim rsID As ADODB.Recordset

```

```

Set rsCancel = New ADODB.Recordset
Set rsID = New ADODB.Recordset

```

```

'Get the most current enrollment.
rsID.Open ("SELECT MAX(pkDetailID) as [pkDetailID] from tblPClubDetail WHERE
accountNumber = " & AccountNumber & ""), db

```

```

'No source available then exit proc.
If IsNull(rsID!pkDetailID) Then Exit Function

```

```

rsCancel.Open ("SELECT fkSourceID from tblPClubDetail WHERE pkDetailID = " &
rsID!pkDetailID & ""), db

```

```

rsID.Close

```

```

frmPClub.cboSource.ListIndex = rsCancel!fkSourceID - 1

```

```

rsCancel.Close

```

```

Exit Function

```

```

EXIT_PROC:
Exit Function

```

```

LoadErr:
MsgBox "Error number: " & Err.Number & vbCrLf & Err.Description
Resume EXIT_PROC

```

```

End Function

```

```

-----
Sub: CancelPCLub
Created on: 02/15/2000
Programmer: Paul Patterson
Purpose: Process to cancel PClub enrollment.
PRE:
POST:
-----

```

```

Public Function CancelPCLub(ByVal AcctNumber As String) As Boolean
On Error GoTo CancelPCLubError

```

```

Dim strMsg As String
Dim intRow As Integer

```

```

CancelPCLub = False

```

```

Call frmPCLub.UpdatePCLubForm(frmPCLub.imgWait, "Cancelling enrollment, please wait...")

```

```

'Verify the connection status.

```

```

If objFDR.FDRGetConnectionStatus = 0 Then

```

```

    MsgBox "Unable to sign on to the FDR system. Please close the FDR session and
    retry.", vbCritical
    frmPCLub.lblMessage = "Unable to sign on to the FDR system."
    frmPCLub.MaskEdAcctNum.SetFocus
    Screen.MousePointer = vbDefault
    Exit Function

```

```

End If

```

```

'Keystrokes to enroll customer.

```

```

objFDR.FDRSendKeys "<Home>"

```

```

objFDR.FDRSendKeys "FPX " & AcctNumber

```

```

objFDR.FDRSendKeys "<EraseEOF>"

```

```

objFDR.FDRSendKeys "<Enter>"

```

```

If objFDR.FDRWatchFPX(TIMEOUT) = 0 Then

```

```

    'Find which row is the PClub fee and delete it.
    intRow = 11 'Starting row.

```

```

    strMsg = Trim(objFDR.FDRGetString(intRow, 15, 8))

```

```

    If Trim(strMsg) <> "" Then

```

```

        Do

```

```

            'Read the fee program field for the current row.

```

```

            strMsg = Trim(objFDR.FDRGetString(intRow, 15, 8))

```

```

            If strMsg = "PCLUB395" Or strMsg = "SECPCLUB" Or strMsg = "FSBCLUB " Or strMsg =
            "CLUBFNC" Then

```

```

                objFDR.FDRMoveTo intRow, 3

```

```

                objFDR.FDRSendKeys "D"

```

```

                objFDR.FDRSendKeys "<PF2>"

```

```

            If objFDR.FDRWatchNMCSVerifyDelete(TIMEOUT) = 0 Then

```

```

                objFDR.FDRSendKeys "<PF11>"

```

```

                If objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0 Then

```

```

                    'Watch for successful screen navigation...

```

```

                    'Switch to NM CS screen and update.

```

```

                    objFDR.FDRSendKeys "<Home>"

```

```

                    objFDR.FDRSendKeys "NM CS"

```

```

                    objFDR.FDRSendKeys "<Enter>"

```

```

                    If objFDR.FDRWatchNMCS(TIMEOUT) = 0 Then

```

```

                        objFDR.FDRMoveTo 10, 50

```

```

                        objFDR.FDRSendKeys " "

```

```

objFDR.FDRSendKeys "<Enter>"

If objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) = 0 Then
    'Update the processed flag for this account.
    db.Execute "UPDATE tblPClub SET enrollFlag = '0' WHERE
AccountNumber = '" & AcctNumber & "'"
    db.Execute "INSERT INTO tblPClubCancel
(accountNumber,userID,cancelDate) VALUES('" & AcctNumber & "','" &
userID & "','" & Date & "'"
    Dim rst As ADODB.Recordset
    Set rst = New ADODB.Recordset

    rst.Open "SELECT @@IDENTITY AS 'Identity'", db
    Call frmPClub.SaveCancelReason(rst!identity)
    rst.Close
    Exit Do

Else
    strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
    Call AppendError(AcctNumber, strMsg)
    Exit Do
End If 'objFDR.FDRWatchNMCSUpdateSuccessful(TIMEOUT) <> 0

Else
    strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
    Call AppendError(AcctNumber, strMsg)
    Exit Do
End If 'objFDR.FDRWatchScreen("CUSTOMER SERVICE", 2, 27, TIMEOUT) = 0

Else
    strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
    Call AppendError(AcctNumber, strMsg)
    Exit Do
End If 'objFDR.FDRWatchFPXUpdateSuccessful(TIMEOUT) = 0

Else
    strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
    Call AppendError(AcctNumber, strMsg)
    Exit Do
End If 'objFDR.FDRWatchNMCSVerifyDelete(TIMEOUT) = 0

Else
    intRow = intRow + 2
End If

Loop Until strMsg = ""

End If 'null test

Else
    strMsg = Trim(objFDR.FDRGetString(24, 1, 75))
    Call AppendError(AcctNumber, strMsg)
    CancelPClub = False
    Exit Function

End If 'objFDR.FDRWatchFPX(TIMEOUT)' = 0

CancelPClub = True
objFDR.FDRSendKeys "<Home>"
objFDR.FDRSendKeys "<PF3>"
Call frmPClub.UpdatePClubForm(frmPClub.imgNotEnrolled, "Cancellation process is complete.")
Beep
Exit Function

EXIT_PROC:
Exit Function

CancelPClubError:
CancelPClub = False
Call AppendError(AcctNumber, Left(Err.Description, 75))
Resume EXIT_PROC
End Function

```

FDR Error codes

**** OPERATOR NOT SIGNED ON ****

You have not successfully signed on to the System, and you are attempting to access a transaction. Use the FDSN transaction to sign on to the System, verifying that you are entering correct sign-on information. Then re-enter the transaction you are attempting to access.

**** PASSWORD WILL EXPIRE IN XX DAYS ****

Your password expires in the number of days indicated. Change your password before it expires.

**** RESTRICTED TERMINAL ****

You have entered a sign-on transaction for which you have no access. Be sure you entered the correct sign-on transaction. If you receive this message after entering a correct sign-on transaction, contact your Security Administrator.

**** SHELL LIFESPAN WILL BE EXCEEDED IN XX DAYS ****

The System purges your shell name in the number of days indicated. Contact your Security Administrator.

**** SIGN OFF IS COMPLETE ****

Your sign-off was successful.

**** SIGN ON IS COMPLETE ****

Your sign-on was successful.

FDRSNP-25 INVALID SIGNON REGION TRY REGION XX

The entry region on your terminal control block definition is incorrect. Call the FDR Response Center for assistance.

FDRSNP-27 PROFILE TABLE EMPTY. . .

A System problem exists. Call the FDR Response Center for assistance.

FDRSNP-28 PROFILE FILE ERROR. . .XX

A System problem exists. Call the FDR Response Center for assistance.

FDRSNPE-04 NO CLIENT SYSTEM RECORD FOR GRID

You have signed on to a nonmaster terminal with a new group identification before the master password has been established. Contact your Security Administrator.

FDRSNPE-05 SECURITY FILE CLOSED. WAIT 2 MIN. AND TRY AGAIN

The Security File is closed for maintenance. If you are unable to sign on after 10 minutes, call the FDR Response Center for assistance.

FDRSNPE-06 SECURITY FILE IO ERROR. NOTIFY SCC

A System problem exists. Call the FDR Response Center for assistance.

FDRSNPE-07 SHELL ALREADY IN USE BY TERMINAL XXXX

Shell has not logged off in core memory. Call the FDR Response Center for assistance.

FDRSNPE-08 TERMINAL PASSWORD DISABLED

You have made three unsuccessful attempts to sign on. If you receive this message at a nonmaster terminal, call your Security Administrator. If you receive this message at a master terminal, call the FDR Response Center for assistance.

FDRSNPE-09 PASSWORD HAS EXPIRED

Your password has expired. You must change your password.

FDRSNPE-11 PASSWORDS DO NOT MATCH. TRY AGAIN

The password you selected does not match your confirmed entry. Try again.

FDRSNPE-12 INVALID PASSWORD ENTERED. TRY AGAIN

You entered a password that does not match the password you previously entered. Re-enter the correct password. If you continue to receive this message, contact your Security Administrator.

FDRSNPE-13 INVALID NAME ENTERED. TRY AGAIN

The shell name you entered is not on file. Check your entry. If you have correctly entered the shell name, call your Security Administrator.

FDRSNPE-14 PREVIOUS PASSWORD AND NEW PASSWORD ARE EQUAL. TRY AGAIN.

The password you entered is the same as your previous password. Enter a different password.

FDRSNPE-15 MASTER NAME INVALID FOR THIS TERMINAL

You entered a master shell name on a nonmaster terminal. Use the master terminal or enter a nonmaster shell name.

FDRSNPE-17 SECURITY FILE UNAVAILABLE FOR READ ONLY. THE MRO LINK IS DOWN.

A System problem exists. Call the FDR Response Center for assistance.

FDRSNPE-18 OPERATOR PROFILE ID XXXXXXXXX NOT FOUND

Your operator profile has not been defined. Call your Security Administrator.

FDRSNPE-19 SIGN ON NOT AVAILABLE AT XX:XXXX

You may not access the System at this time.

FDRSNPE-25 SIGNON NOT AVAILABLE ON XXXXXXDAY

You may not access the System on the specified day.

FDRSNPE-26 THE MINIMUM PASSWORD LENGTH X

The password you enter must be at least the specified length.

FDRSNPE-27 PASSWORD HAS BEEN ACCEPTED. SIGN ON IS COMPLETE

You have successfully established your password and your sign on was successful.

FDRSNPE-28 CURRENT PASSWORD AND NEW PASSWORD ARE EQUAL. TRY AGAIN

You entered your current password for the new password. Enter a different password.

FDRSNPE-29 ILLEGAL PASSWORD CHOSEN. TRY ANOTHER

The System does not accept the password you chose. Enter a different password.

FDRSNPE-30 SHELL DISABLED

You have made three unsuccessful attempts to sign on, and the System has disabled your shell. Call your Security Administrator.

FDRSNPE-31 PASSWORD CHOSEN TOO SIMILAR TO SHELL NAME

The shell name may not be wholly contained in the password. Enter a different password.

SNP-01 INVALID TRANSACTION CODE

Call the FDR Response Center for assistance.

SNP-03 TCB FILE CLOSED. WAIT 2 MIN AND TRY AGAIN.

The Terminal Control Block file is closed for maintenance. If you are unable to sign on after 10 minutes, call the FDR Response Center for assistance.

SNP-04 TCB FILE ERROR. NOTIFY SCC.

A System problem exists. Call the FDR Response Center for assistance.

SNP-05 PASSWORD FILE CLOSED. WAIT 2 MIN AND TRY AGAIN.

The password file is closed for maintenance. If you are unable to sign on after 10 minutes, call the FDR Response Center for assistance.

SNP-06 PASSWORD FILE IO ERROR. NOTIFY SCC.

A System problem exists. Call the FDR Response Center for assistance.

SNP-15 SHELL HAS EXCEEDED ITS LIFESPAN

The shell name has expired and been deactivated. Contact your Security Administrator. The Security Administrator may delete and re-add your shell name or change the lifespan option on the shell.

SNP-18 TERMINAL PROFILE ID XXXXXXXXX NOT FOUND

Your terminal profile has not been defined. Call your Security Administrator.

SNP-19 INVALID INPUT TRY AGAIN

You entered invalid characters. Re-enter valid characters. If you continue to receive the message, call the FDR Response Center for assistance.

INDEX

C

cancellation, 9, 10, 34, 35, 61, 64
CAPITAL Card Services, 4
CAPS, 5, 6, 7, 33, 57
CCSI, 4, 5, 12, 22
Collections, 4, 24
credit line, 5, 6
CSI, 5, 9, 11, 21, 25, 28, 48
Customer Service, 4, 5

D

Database Design Language, 2, 17

E

enrollment, 5, 6, 9, 10, 28, 29, 35, 36, 44, 48, 51, 69, 72, 73

F

FDR, 5, 7, 22, 23, 24, 63, 64, 68, 73, 75, 76. *See* First Data Resources
FPX, 3, 7, 8, 9, 45, 49, 50, 53, 54, 64, 68, 69, 73

G

Gantt Chart, 3, 11

I

Incentive Report, 2, 25

IVR, 6, 25, 48
IWR, 5

L

Logical Database Design, 2, 14

M

Marketing, 4, 21

N

Network Architecture, 12
New Accounts, 4
NM CS, 8, 45, 49, 53, 69, 73
NM NA, 9, 49, 53

P

Physical Database Design, 2, 15
PREMIUM Club, 5, 6
Process Modeling, 2, 13
Project Plan, 2, 11

R

report, 10, 25, 26, 27, 61
Reports, 2, 3, 10, 25
Requirements, 2, 6, 11
Roborep, 2, 5, 9, 11, 42

Implementation of Premium Club at CAPITAL Card Services

By
Paul Patterson

Introduction

- CAPITAL Card Services
 - CAPITAL is a credit card servicing company.
 - It delivers all the components necessary to execute the card issuing business plan.
 - It provides the infrastructure and human capital needed to support a growing card business.

Sub-Prime

- The sub-prime product:
 - Limited credit lines
 - Fees priced for risk
 - Designed to re-establish credit
- The sub-prime customer:
 - Damaged, poor or limited credit history
 - In need of credit education
 - High transactions on accounts

Strategic Business Alliances

- First Data Resources (FDR)
- Central States Indemnity
- Intelligent Analytics (IA)
- US Marketing Corporation (USM)
- DIMAC Direct

What is Premium Club?

- A membership program in which members are eligible for products and services designed to save them money.
 - Car rental savings
 - Premium Protection for credit cards
 - 48 hour-emergency card replacement
 - Savings on flower and gift basket orders
 - Financial management guide
 - Emergency cash
 - Frequent credit line increase reviews

Project Plan

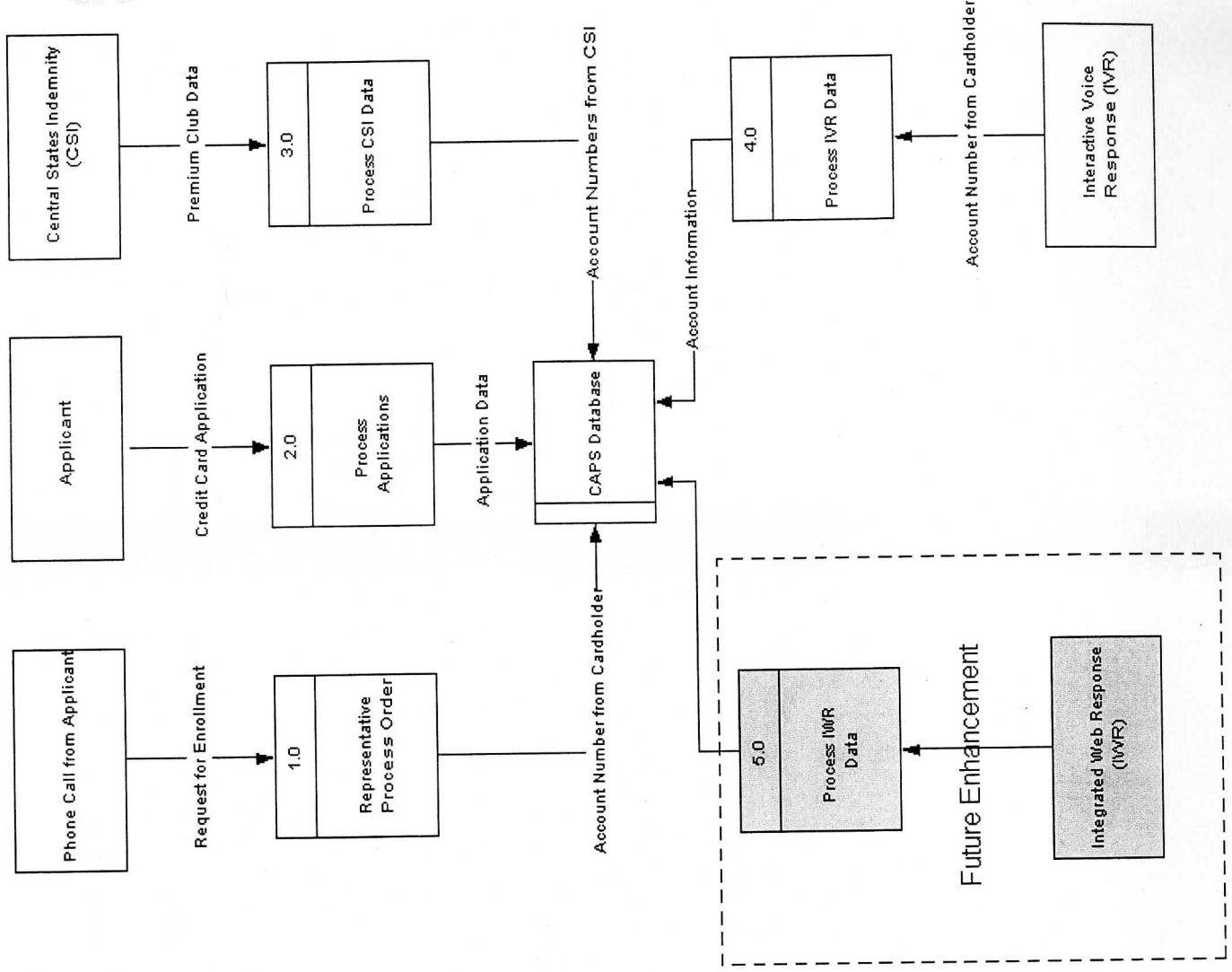
Table 1 Task Information

| ID | Task_Name | Duration | Pred. | Start_Date | Finish_Date | Slack |
|----|------------------------------------|----------|-------|------------|-------------|---------|
| 1 | Requirements Collection | 15 days | | 01-Dec-00 | 21-Dec-00 | 0 days |
| 2 | Process Design | 5 days | 1 | 22-Dec-00 | 28-Dec-00 | 0 days |
| 3 | Database Design | 5 days | 2 | 29-Dec-00 | 04-Jan-01 | 5 days |
| 4 | CAPS Modifications | 20 days | 2 | 29-Dec-00 | 25-Jan-01 | 0 days |
| 5 | Incorporate Host Connectivity | 20 days | 1 | 29-Dec-00 | 25-Jan-01 | 0 days |
| 6 | Design Enrollment Form | 10 days | 3 | 05-Jan-01 | 18-Jan-01 | 5 days |
| 7 | Roborep Modifications | 8 days | 2 | 29-Dec-00 | 09-Jan-01 | 12 days |
| 8 | Create Batch Processing Procedures | 5 days | 2 | 29-Dec-00 | 04-Jan-01 | 15 days |
| 9 | Automate IVR File | 3 days | 3 | 05-Jan-01 | 09-Jan-01 | 12 days |
| 10 | Automate CSI File | 3 days | 3 | 05-Jan-01 | 09-Jan-01 | 12 days |
| 11 | Automate Label Printing | 2 days | 3 | 05-Jan-01 | 08-Jan-01 | 13 days |
| 12 | Report Design | 5 days | 4,7 | 26-Jan-01 | 01-Feb-01 | 10 days |
| 13 | Convert Data from Legacy System | 2 days | 4,7 | 26-Jan-01 | 29-Jan-01 | 13 days |
| 14 | Testing | 15 days | 4,7 | 26-Jan-01 | 15-Feb-01 | 0 days |

Data Flow Diagram

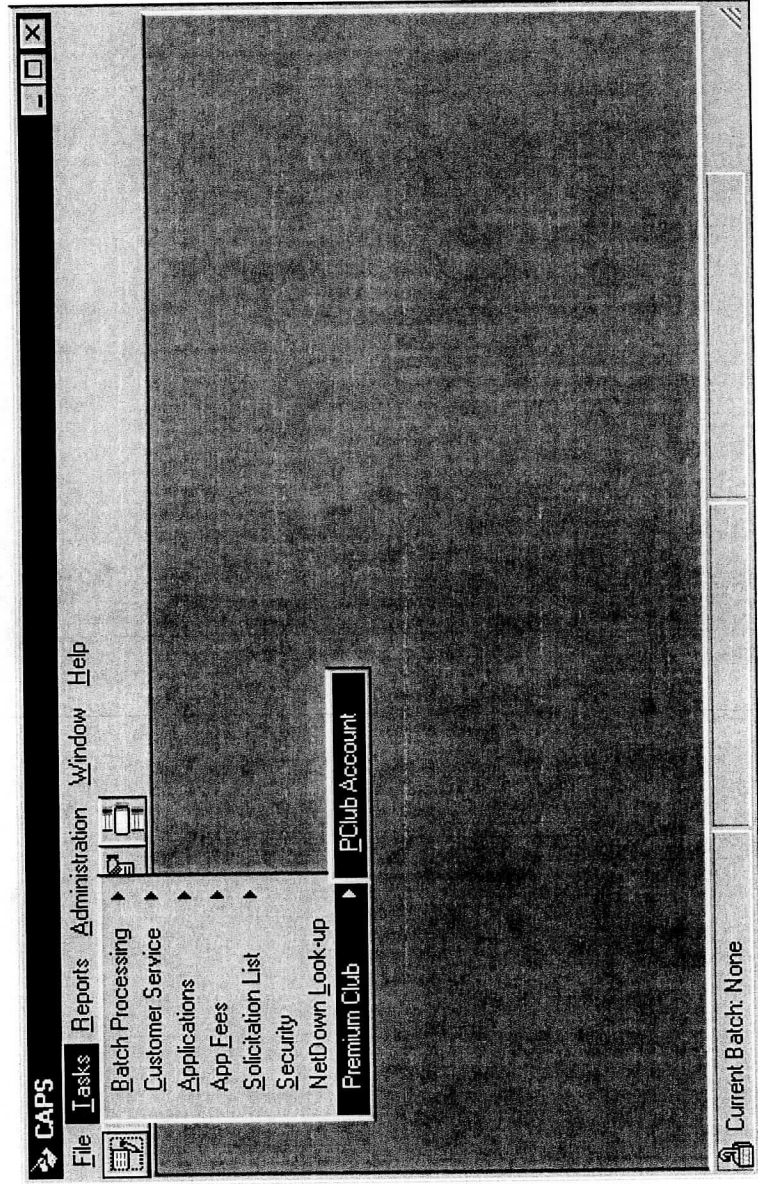
- Enrollment Sources

- Phone-In
- Application
- CSI
- IVR
- IWR



- CAPITAL Application Processing System
 - Enroll cardholders real-time

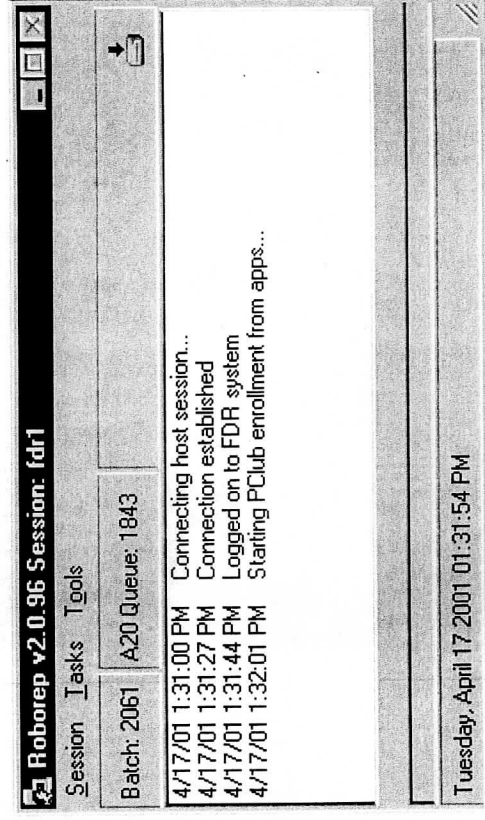
CAPS



CAPITAL
CARD SERVICES INC.

Roborep

- Terminal Emulator
 - HLLAPI Protocol
 - Connects to a mainframe, logs in and initiates various functions on the mainframe
 - "Screen Scraping"



Premium Club Form

Account Number

4239-8010-0105-7856

Account Information

Status:

Please wait while FDR connection is established...

☐ Enrolled in Premium Club

Date Enrolled

Open Date

Last CLI Date

Source

☐ Cancel Premium Club

Last Billing Date

Next Billing Date

Date Canceled

Reason(s) for Canceling

Processing

Close

Servicing

No CLI Given

No Packet Received

Coupons not Valuable

Premium Club Form

CAPITAL
CARD SERVICES INC.

| Premium Club | |
|--|---|
| Account Number 4239-8010-0105-7864 | Account Information ACCOUNT,TEST 500 E 60TH ST N C/O CAPITAL CAR SIOUX FALLS SD 57104-0478 |
| <div>Status: This account is enrolled in PClub.</div> | |
| <input checked="" type="checkbox"/> Enrolled in Premium Club | Date Enrolled 04/26/01 |
| <input type="checkbox"/> Cancel Premium Club | Last Billing Date: 00/00/00 |
| | Open Date: 03-00 |
| | Last CLI Date: 11-00 |
| | Source Insert |
| | Reason(s) for Canceling Servicing No CLI Given No Packet Received Coupons not Valuable |
| <div>Process Close</div> | |


Premium Club Form

CAPITAL
CARD SERVICES INC.

| Premium Club | |
|---|--|
| Account Number 4239-8010-0105-7864 | Account Information ACCOUNT TEST |
| | 500 E 60TH ST N C/O CAPITAL CAR |
| | SIOUX FALLS SD 57104-0478 |
| <input type="checkbox"/> Enrolled in Premium Club | Date Enrolled 04/26/01 |
| <input checked="" type="checkbox"/> Cancel Premium Club | Last Billing Date: 00/00/00 |
| | Open Date: 03-00 |
| | Last CLI Date: 11-00 |
| | Source |
| | Reason(s) for Canceling Servicing No CLI Given No Packet Received Coupons not Valuable |
| <div>Status: This account has canceled PClub.</div> | |
| <div>Process Close</div> | |

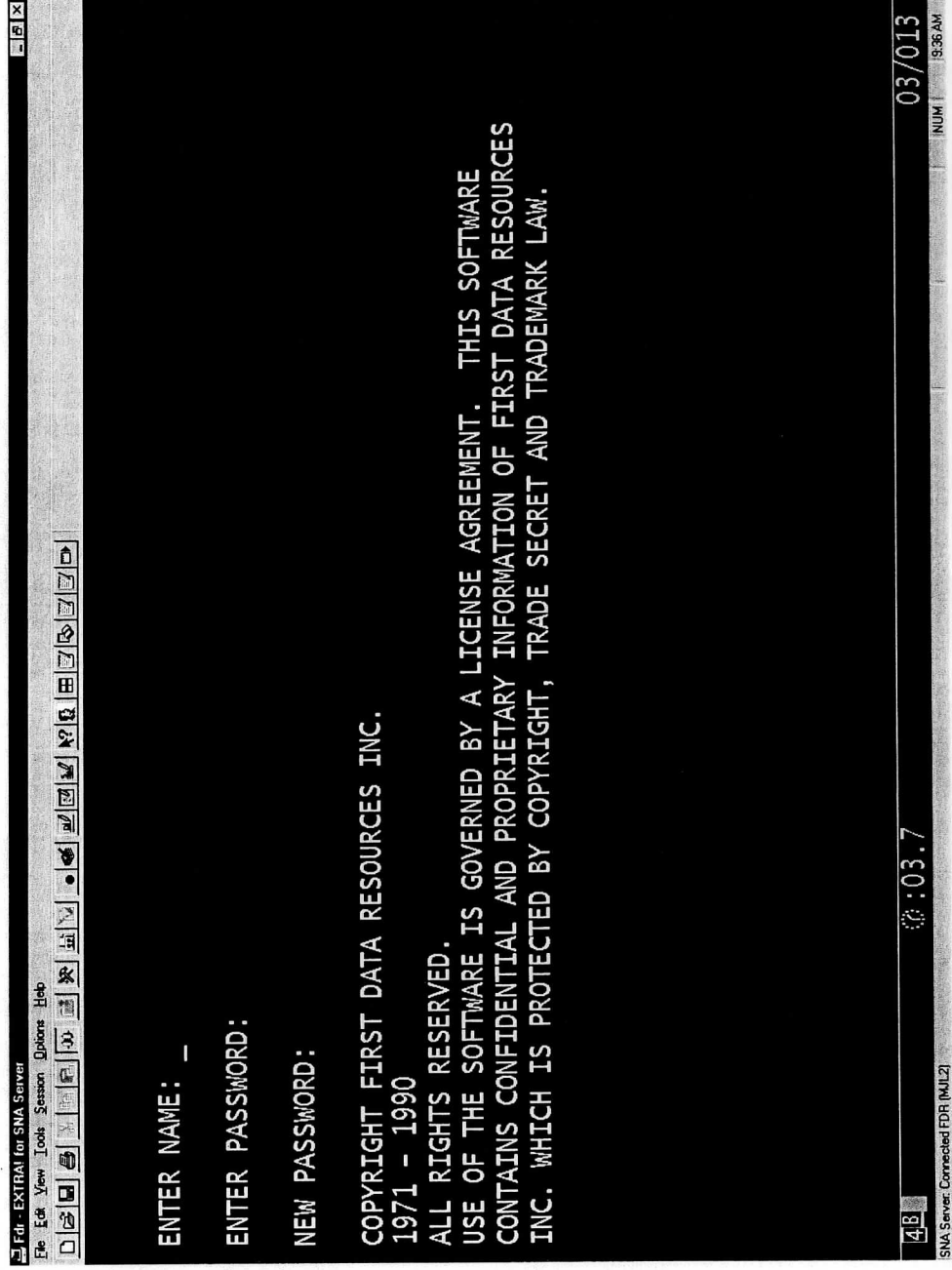
Premium Club Form

CAPITAL
CARD SERVICES INC.

| Premium Club | |
|--|--|
| Account Number | Account Information |
| 4239-8010-0105-7864 | <div> </div> <div> </div> <div> </div> |
| <div>Status:  ** INVALID CARDHOLDER NUMBER **</div> | |
| <input type="checkbox"/> Enrolled in Premium Club | Date Enrolled: <div> </div> Open Date: <div> </div> Last CLI Date: <div> </div> Source: <div> </div> |
| <input type="checkbox"/> Cancel Premium Club | Last Billing Date: <div> </div> Next Billing Date: <div> </div> Date Canceled: <div> </div> Reason(s) for Canceling: <div>Servicing No CLI Given No Packet Received Coupons not Valuable</div> |
| <div>Process</div> <div>Close</div> | |

Enrollment Process

- Step 1:
Logon



Enrollment Process

- Step 2:
Verify
Logon is
successful



- ```

CIS 4239801001057864
ACCOUNT,TEST*COLLECTIONS*500 E 60TH ST N*C/O CAPITAL CARD SERVICES*SIOUX FALLS*S
D*57104-0478*4239801001057864
046 041301 N MJK1009 NM-022 = 10; PREV CYCLE = 20
045 041301 N MJK1009 NM-016 = ; PREV STATUS = C
044 041301 J 1536 MJK1009 TC253 BTID LF AM00000000002000 AM2000000000000000C
043 041301 J 1535 MJK1009 TC253 BTID PN AM00000000011900 AM2000000000000000C
042 CCI 013001 C 1832 RQST DUDT CHNGD TO 0202
041 012501 N PHYHXXW NM-016 = C; PREV STATUS =
040 012501 N PHYGAXB NM-016 = ; PREV STATUS = C
039 012501 N PHYGAXB NM-016 = C; PREV STATUS = C
038 012501 N MJMQTXV NM-016 = C; PREV STATUS =
037 ZMW 012501 C 1438 VAIG/GAV ALL OPTIONS DIDN'T AGRE WITH THEM 01/25/01
036 RFD 012501 C 1434 CH HAD BETTER TERMS ELSEWHERE
035 012501 J 1433 PHX9MXP TC255 BTID AF AM000000000002500 AM2000000000000000C
034 012501 J 1431 MJMQTXV TC255 BTID AF AM000000000002500 AM2000000000000000C
033 TLV 012501 C 1429 CRDT 1/2 THE A/F
032 ZMW 012501 C 1427 VAIG/EXP HOW PROCES FEE WORK AND WHAT THY ARE 01/25/01
031 TLV 012501 C 1426 GV CM LWR INT RT AT 149 FOR 6 MO.
030 ARB 012501 C 1426 CCI VAIG REQ TO CLS ACCT NO LNGR NEEDED
029 RFD 012501 C 1425 CH ACCT WAS CLOSED AT THIS TIME AS 1X CRTS
028 ARB 012501 C 1422 VAIG CCI REQ FEE REV ED CH 1X CRTSY

TRANSACTION COMPLETED

```

# Enrollment Process

- Step 4:  
Access the  
Fees  
Screen

EXTRA! for SNA Server  
File Edit View Tools Session Options Help

@FPX 4239801001057864  
@FPX  
CLIENT 5386  
ACCT NO: ACCOUNT, TEST  
ACCOUNT LEVEL FEES  
04/26/01 09:40  
PAGE 1

| DELETE    | FEE      | GROUP     | ACCOUNT         | ASMT  | FEE DATE | RECORD DATE |
|-----------|----------|-----------|-----------------|-------|----------|-------------|
| ACTN DATE | PROGRAM  | ID        | FEE AMT         | TIMES | NEXT     | RGSTR       |
|           |          |           |                 | COUNT | LAST     | UPDATE      |
| -         | FN-FPERR | 999999999 | 999999999999.99 | 999   | 05/16/01 | 02/16/01    |
|           |          |           |                 | 000   | 04/16/01 | 04/16/01    |

PF2=UPDATE PF3=EXIT PF6=GROUP LIST PF7=BKWD PF8=FWD  
PF9=VIEW PRG PF10=LIST PRGMS PF11=VERIFY DELETE PF12=CANCEL UPDATE  
NO MORE DATA V2FPACCT 08  
11/003

0:02.2  
SNA Server Connected FDR (MUL2) NUM 19:40 AM

# Enrollment Process

- Step 5:  
Add the  
new fee

EDIT - EXTRAL for SNA Server  
File Edit View Tools Session Options Help

@FPX 4239801001057864  
CLIENT 5386  
ACCOUNT LEVEL FEES  
04/26/01 09:40  
PAGE 1

ACCT NO: ACCOUNT, TEST

| DELETE    | FEE      | GROUP     | ACCOUNT         | ASMT  | FEE DATE | RECORD DATE |
|-----------|----------|-----------|-----------------|-------|----------|-------------|
| ACTN DATE | PROGRAM  | ID        | FEE AMT         | TIMES | NEXT     | RGSTR       |
|           |          |           |                 | COUNT | LAST     | UPDATE      |
|           | FN-FPERR | 999999999 | 999999999999.99 | 999   | 05/16/01 | 02/16/01    |
|           |          |           |                 | 000   | 04/16/01 | 04/16/01    |
| A         | PCLUB395 |           | -               |       |          |             |

PF2-UPDATE PF3=EXIT PF6=GROUP LIST PF7=BKWD PF8=FWD  
PF9=VIEW PRG PF10=LIST PRGMS PF11=VERIFY DELETE PF12=CANCEL UPDATE  
NO MORE DATA V2FPACCT 08

13/038  
:02.2  
SNA Server Connected FDR (MIL2) NUM 1 18:41 AM

CAPITAL  
CARD SERVICES INC

# Enrollment Process

- Step 6:  
Verify the  
update was  
successful

FDI - EXTRAI for SNA Server  
File Edit View Tools Session Options Help

ACCOUNT LEVEL FEES

@FPX 4239801001057864  
@FPX  
CLIENT 5386  
ACCT NO: ACCOUNT,TEST

04/26/01 09:40  
PAGE 1

DELETE FEE GROUP  
ACTN DATE PROGRAM ID

FN-FPERR 99999999  
PCLUB395 00002000

ACCOUNT ASMT FEE DATE  
FEE AMT TIMES NEXT LAST  
999999999999.99 999 05/16/01  
999999999999.99 999 04/16/01  
999999999999.99 999 04/30/01  
999999999999.99 999 00/00/00

RECORD DATE  
RGSTR UPDATE  
02/16/01  
04/16/01  
04/26/01  
04/26/01

PF2=UPDATE PF3=EXIT PF6=GROUP LIST PF7=BKWD PF8=FWD  
PF9=VIEW PRG PF10=LIST PRGMS PF11=VERIFY DELETE PF12=CANCEL UPDATE  
UPDATE SUCCESSFUL

4B :03.0  
11/003  
SNA Server Connected FDI (MULTI) NUM 1941 AM

# Enrollment Process

- Step 7:  
Access the  
customer  
service  
screen

**\*NM**

**CS**      **CUSTOMER SERVICE**      **CRCD: 840**      **04/26/01 09:43**

**ACCOUNT: 4239801001057864**      **ACCOUNT, TEST**

**PARTNER ID:**      **ACCOUNT, COLLECTIONS**

**REQUEST STATEMENT: (Y) STMT HOLD CODE: USAVE: 000000000 (+/-)**

**BILLING CYCLE CODE: 10 USAVE GROUP ID: Z SORT DISTRIBUTION CODE: 000**

**GET NEW ADDRESS: (Y) OPT-OUT CD: 0 PAY AHEAD FLAG: 0**

**RANDOM DIGITS: 16**      **CUSTOMER INITIATED SKIP PAY: (Y)**

**SPECIAL FLAGS: 5: 6: 7: 8:**      **AFFINITY NUMBER:**

**MISC FIELDS: 1: 2: 3: 714 4: P 5: 6: 7: 8: 9: 10:**

**ISSUER-DEFINED FLAGS: 1: 2: 3: 4:**

**LANG CD: EN SERVICES: 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0**      **NO MAIL FLAG: 0**

**BONUS DOLLAR CREDIT TOTAL: 0000000**      **BONUS DOLLAR DEBIT TOTAL: 0000000**

**BONUS DOLLARS: CREDIT/DEBIT: (C/D) AMOUNT: TOTAL: 000000000**

**QUALIFIED PURCHASES: CREDIT/DEBIT: (R/S/X/Y) AMOUNT: TOTAL: 000000000**

**SKIP-PAY ELIGIBILITY IND: Y**      **SELECTED DISBURSEMENT METHOD:**

**CURRENT ANNIV. REBATE: CREDIT/DEBIT: (E/F) AMOUNT: TOTAL: 0000000**

**LAST ANNIV. REBATE: CREDIT/DEBIT: (G/H) AMOUNT: TOTAL: 0000000**

**COMPUTER LTR NUMBER: PENDING LTR REQUESTS TODAY:**

**VARIABLE #1: VARIABLE #2:**

**VARIABLE #3: VARIABLE #4:**

**SKIP PAY HISTORY: AFLT INSRNC CD: N BAGG INSRNC CD: N**

**ENTER ACCOUNT RELATED CHANGES**

**4B**      **:01.3**      **10/050**

SNA Server Connected FOR (MIL2)      NUM      9:43 AM

- Step 8:  
Verify  
changes  
were  
successful

# Enrollment Process

File Edit View Tools Session Options Help

CS\_\*NM CUSTOMER SERVICE CRCD: 840 04/26/01 09:43

ACCOUNT: 4239801001057864 ACCOUNT, TEST

PARTNER ID: ACCOUNT, COLLECTIONS

REQUEST STATEMENT: (Y) STMT HOLD CODE: USAVE: 000000000 (+/-)

BILLING CYCLE CODE: 10 USAVE GROUP ID: Z SORT DISTRIBUTION CODE: 000

GET NEW ADDRESS: (Y) OPT-OUT CD: 0 PAY AHEAD FLAG: 0

RANDOM DIGITS: 16 CUSTOMER INITIATED SKIP PAY: (Y)

SPECIAL FLAGS: 5: 6: 7: 8: AFFINITY NUMBER:

MISC FIELDS: 1: 2: 3: 714 4: P 5: 10:

6: 7: 8: 9:

ISSUER-DEFINED FLAGS: 1: 2: 3: 4: NO MAIL FLAG: 0

LANG CD: EN SERVICES: 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 BONUS DOLLAR DEBIT TOTAL: 0000000

BONUS DOLLAR CREDIT TOTAL: 0000000 BONUS DOLLAR DEBIT TOTAL: 000000000

BONUS DOLLARS: CREDIT/DEBIT: (C/D) AMOUNT: TOTAL: 000000000

QUALIFIED PURCHASES: CREDIT/DEBIT: (R/S/X/Y) AMOUNT: TOTAL: 000000000

SKIP-PAY ELIGIBILITY IND: Y SELECTED DISBURSEMENT METHOD:

CURRENT ANNIV. REBATE: CREDIT/DEBIT: (E/F) AMOUNT: TOTAL: 0000000

LAST ANNIV. REBATE: CREDIT/DEBIT: (G/H) AMOUNT: TOTAL: 0000000

COMPUTER LTR NUMBER: PENDING LTR REQUESTS TODAY:

VARIABLE #1: VARIABLE #2:

VARIABLE #3: VARIABLE #4:

SKIP PAY HISTORY: AFLT INSRNC CD: N BAGG INSRNC CD: N VSQNMTRN 0E

TRANSACTION COMPLETED SUCCESSFULLY

01/003

SNA Server, Connected FDR (MUL2) CAP NUM 19:43 AM



# Enrollment Process

- Step 9:  
Access  
account  
information  
screen

FDI - EXTERNAL for SNA Server  
File Edit View Tools Session Options Help

NAME/ADDRESS CHANGES 04/26/01 09:48

NA  
ACCOUNT: 4239801001057864 ACCOUNT, TEST  
ACCOUNT, COLLECTIONS

PRINCIPAL CARDHOLDER NAME: ACCOUNT, TEST LETTER CATEGORY GRPS:  
SECONDARY CARDHOLDER NAME: ACCOUNT, COLLECTIONS A B C D E F  
SALUTATION CODE: U G H I J K L  
HOME PHONE NUMBER: 0000000000 M N O P Q T  
BUSINESS PHONE NUMBER: 0000000000 U V Z  
ADDRESS LINE 1: 500 E 60TH ST N LINE 2: C/O CAPITAL CARD SERVICES  
CITY: SIOUX FALLS STATE: SD ZIP CODE: 571040478  
ADDRESS CHANGE LETTER: PROPAGATE:  
PENDING LETTER REQUESTS TODAY:  
COMPUTER LETTER NUMBER:  
VARIABLE #1:  
VARIABLE #2:  
VARIABLE #3:  
VARIABLE #4:

LANG CD: EN  
FORCE EMBOSS: NAME-TO-EMBOSS CODE: NUMBER OF PLASTICS: CARD NO:  
NAME-TO-BE EMBOSSED: RUSH MAIL:  
PLASTIC REPLACEMENT FEE:  
PF-4 EXPANDED ADDRESS MAINTENANCE  
ENTER ACCOUNT RELATED CHANGES

4.9 :02.0 05/028  
SNA Server: Connected FDR (MILL) NUM 9:48 AM

# Cancellation Process

- Step 1:  
Access the  
fees screen

Fdr - EXTRAL for SNA Server  
File Edit View Tools Session Options Help

@FPX 4239801001057864  
@FPX  
CLIENT 5386  
ACCT NO: ACCOUNT,TEST  
04/26/01 10:13  
PAGE 1

ACCOUNT LEVEL FEES

| DELETE    | FEE      | GROUP     | ACCOUNT         | ASMT  | FEE DATE | RECORD DATE |
|-----------|----------|-----------|-----------------|-------|----------|-------------|
| ACTN DATE | PROGRAM  | ID        | FEE AMT         | TIMES | NEXT     | RGSTR       |
|           |          |           |                 | COUNT | LAST     | UPDATE      |
|           | FN-FPERR | 999999999 | 999999999999.99 | 999   | 05/16/01 | 02/16/01    |
|           |          |           |                 | 000   | 04/16/01 | 04/16/01    |
| D         | PCLUB395 | 00002000  | 999999999999.99 | 999   | 04/30/01 | 04/26/01    |
|           |          |           |                 | 000   | 00/00/00 | 04/26/01    |

PF2=UPDATE PF3=EXIT PF6=GROUP LIST PF7=BKWD PF8=FWD  
PF9=VIEW PRG PF10=LIST PRGMS PF11=VERIFY DELETE PF12=CANCEL UPDATE  
NO MORE DATA V2FPACCT 08  
13/038

SNA Server: Connected FDR (MUL2)  
:02.4  
NUM 1014AM

# Cancellation Process

- Step 2:  
Verify the  
cancel was  
successful

FDI - EXTRA! for SNA Server  
File Edit View Tools Session Options Help

@FPX 4239801001057864  
@FPX  
CLIENT 5386

ACCOUNT LEVEL FEES

ACCT NO: ACCOUNT, TEST

| DELETE    | FEE      | GROUP     | ACCOUNT         | ASMT  | FEE DATE | RECORD DATE |
|-----------|----------|-----------|-----------------|-------|----------|-------------|
| ACTN DATE | PROGRAM  | ID        | FEE AMT         | TIMES | NEXT     | RGSTR       |
|           |          |           |                 | COUNT | LAST     | UPDATE      |
| -         | FN-FPERR | 999999999 | 999999999999.99 | 999   | 05/16/01 | 02/16/01    |
|           |          |           |                 | 000   | 04/16/01 | 04/16/01    |
| 04/26/01  | PCLUB395 | 00002000  | 999999999999.99 | 999   | 99/99/99 | 04/26/01    |
|           |          |           |                 | 000   | 00/00/00 | 04/26/01    |

PF2=UPDATE PF3=EXIT PF6=GROUP LIST PF7=BKWD PF8=FWD  
PF9=VIEW PRG PF10=LIST PRGMS PF11=VERIFY DELETE PF12=CANCEL UPDATE  
UPDATE SUCCESSFUL V2FPACCT 0A

11/003  
SNA Server: Connected FDI (MILLZ) 10:16 AM

# Cancellation Process

- Step 3:  
Access the  
customer  
service  
screen

**\*NM**

**CS** CUSTOMER SERVICE CRCD: 840 04/26/01 09:43

ACCOUNT: 4239801001057864 ACCOUNT, TEST

PARTNER ID: ACCOUNT, COLLECTIONS

REQUEST STATEMENT: (Y) STMT HOLD CODE: USAVE: 000000000 (+/-)

BILLING CYCLE CODE: 10 USAVE GROUP ID: Z SORT DISTRIBUTION CODE: 000

GET NEW ADDRESS: (Y) OPT-OUT CD: 0 PAY AHEAD FLAG: 0

RANDOM DIGITS: 16 CUSTOMER INITIATED SKIP PAY: (Y)

SPECIAL FLAGS: 5: 6: 7: 8: AFFINITY NUMBER:

MISC FIELDS: 1: 2: 3: 714 4: P 5: 6: 7: 8: 9: 10:

ISSUER-DEFINED FLAGS: 1: 2: 3: 4: NO MAIL FLAG: 0

LANG CD: EN SERVICES: 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 7: 0 BONUS DOLLAR DEBIT TOTAL: 00000000

BONUS DOLLAR CREDIT TOTAL: 00000000 BONUS DOLLAR DEBIT TOTAL: 00000000

BONUS DOLLARS: CREDIT/DEBIT: (C/D) AMOUNT: TOTAL: 0000000000

QUALIFIED PURCHASES: CREDIT/DEBIT: (R/S/X/Y) AMOUNT: TOTAL: 0000000000

SKIP-PAY ELIGIBILITY IND: Y SELECTED DISBURSEMENT METHOD:

CURRENT ANNIV. REBATE: CREDIT/DEBIT: (E/F) AMOUNT: TOTAL: 00000000

LAST ANNIV. REBATE: CREDIT/DEBIT: (G/H) AMOUNT: TOTAL: 00000000

COMPUTER LTR NUMBER: PENDING LTR REQUESTS TODAY:

VARIABLE #1: VARIABLE #2:

VARIABLE #3: VARIABLE #4:

SKIP PAY HISTORY: AFLT INSRNC CD: N BAGG INSRNC CD: N VSQNMTRN OE

TRANSACTION COMPLETED SUCCESSFULLY

**4B** :01.9 01/003

SNA Server Connected FDR (MIL2) CAP NUM 19:43 AM

# Reports

Table 1 Premium Club Reports

| Report Name  | Description                                                                                                                                                                                                                                              |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Error        | Any account enrollments that are entered, but were not able to be enrolled. Include the source of enrollment.                                                                                                                                            |
| Enrollment   | An enrollment report to determine incentives. The option of day, month and year and the option of choosing the enrollment source and representative who enrolled the cardholder. This should be sortable by enrollment source, month, or representative. |
| Cancellation | A report listing accounts that have canceled with the option of day, month and year and the cancellation reason.                                                                                                                                         |
| Applications | A report showing the number of applications received each month with PClub marked. Identify those applications that are approved.                                                                                                                        |

Any Questions?

**CAPITAL**  
CARD SERVICES INC.